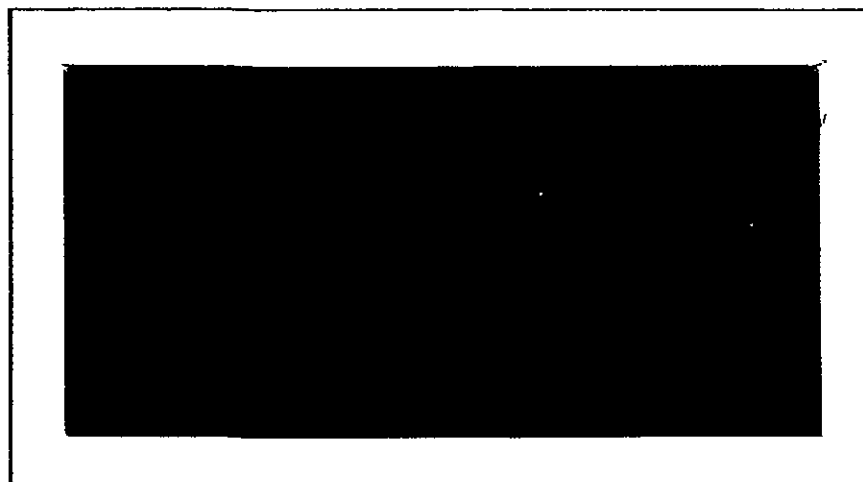


1 up
NASA CR-122332

30



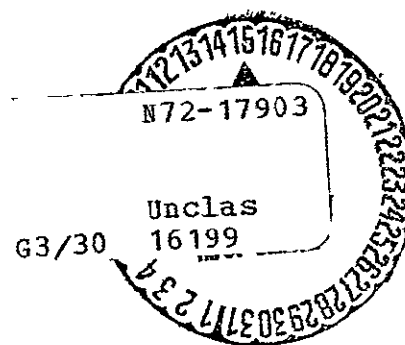
MARTIN MARIETTA

**AEROSPACE
GROUP**

DENVER DIVISION, POST OFFICE BOX 179, DENVER, COLORADO 80201

Reproduced by
NATIONAL TECHNICAL
INFORMATION SERVICE
U S Department of Commerce
Springfield VA 22151

602
(ACCESSION NUMBER)
(NASA-CR-122332) SPACE TRAJECTORIES ERROR
ANALYSIS (STEAP) PROGRAMS. VOLUME 2:
PROGRAMMERS' MANUAL, UPDATE (Martin
Marietta Corp.) Dec. 1971 419 p CSCL 22C
830 p



SPACE TRAJECTORIES ERROR ANALYSIS (STEAP)
PROGRAMS

Volume II - Programmers' Manual (Update)

December 1971

MARTIN MARIETTA CORPORATION
DENVER DIVISION
Denver, Colorado 80201

MCR- 71-4

SPACE TRAJECTORIES
ERROR ANALYSIS PROGRAMS
VERSION II

VOLUME II: PROGRAMMER'S MANUAL

March 1971
Space Navigation Technology
Martin-Marietta Corporation
Denver, Colorado

MCR-71-4

Volume II of Three Volumes

Final Report

Contract NAS 5-11795

Computer Program for Mission Analysis
of Lunar and Interplanetary Missions

Prepared by

E. D. Vogt
G. L. Adams
M. M. Working
J. B. Ferguson
M. R. Bynum

Space Navigation Technology Section
Martin Marietta Corporation
Denver, Colorado
:

For

National Aeronautics and Space Administration
Goddard Space Flight Center
Greenbelt, Maryland

March, 1971

Acknowledgements

The authors wish to acknowledge the invaluable support of several persons whose help was instrumental in the completion of this contract. Dr. Al Bradt, serving as Program Manager, efficiently solved the many administrative problems that arose during the course of the contract. Joanne Spofford did her usual excellent job of typing the voluminous documents. Deborah Bower was responsible for all the graphics work involved with the production of the flowcharts and mathematical analyses. A utility computer program developed by Jim Schnelker aided greatly in the program conversion from CDC single precision to IBM double precision. All of their work was greatly appreciated.

PRECEDING PAGE BLANK NOT FILMED

FOREWORD

STEAP II is a series of three computer programs developed by the Martin Marietta Corporation for the mathematical analysis of interplanetary or lunar navigation and guidance. *STEAP* is an acronym for *Space Trajectory Error Analysis Programs*. The first series of programs under this name was developed under Contract NAS1-9745 for Langley Research Center and was documented in two volumes (*STEAP Users' Manual*, *STEAP Analytical Manual*) as NASA Contract Report 66818. Under contracts NAS5-11795 and NAS5-11873, the STEAP series was extensively modified and expanded for Goddard Space Flight Center. This second-generation series of programs is referred to as STEAP II.

STEAP II is composed of three independent yet related programs: NOMNAL, ERRAN, and SIMUL. All three programs require the integration on n-body trajectories for both interplanetary and lunar missions. The virtual-mass technique is the scheme used for this purpose in all three programs.

The first program named NOMNAL is responsible for the generation of n-body nominal trajectories (either lunar or interplanetary) performing a number of deterministic guidance events. These events include initial or injection targeting, midcourse retargeting, orbit insertion, and miniprobe targeting. A variety of target parameters are available for the targeting events. The actual targeting is done iteratively either by a modified Newton-Raphson algorithm or by a steepest-descent/conjugate gradient scheme. Planar and non-planar strategies are available for the orbit insertion computation. All maneuvers may be executed either by a simple impulsive model or by a pulsing sequence model.

ERRAN, the second program of STEAP II, is used to conduct linear error analysis and generalized covariance analysis studies along specific targeted trajectories. The targeted trajectory may, however, be altered during flight by retargeting events (computed either by linear or nonlinear guidance) and by an orbit insertion event. Knowledge and control covariances are propagated along the trajectory through a series of measurements and guidance events in a totally integrated fashion. The knowledge covariance is processed through measurements using a Kalman-Schmidt or equivalent recursive weighted-least-squares filter with arbitrary solve-for/consider augmentation. Execution of guidance events may be modeled either by an impulsive approximation or by a pulsing sequence model. The resulting knowledge and control covariances can be analyzed by the program at various events to determine statistical data, including

probabilistic midcourse correction sizing and effectiveness, probability of impact, and biased aimpoint requirements. Probe release events are also available for studying missions employing multi-probe spacecraft.

The third and final program in the STEEP II series is the simulation program SIMUL. SIMUL is responsible for the testing of the mathematical models used in the navigation and guidance process. An "actual" dynamic model is used to propagate an "actual" trajectory. Noisy measurements from this "actual" trajectory are then sent to the estimation algorithm. Here the actual measurement, the statistics associated with that measurement, and an "assumed" dynamical model are blended together to generate the filter estimate of the trajectory state. This process is repeated continually through the measurement schedule. At guidance events, corrections are computed based on the estimate of the current state. These corrections are then corrupted by execution errors and added to the "actual" trajectory. The statistics and augmentation of the filter, the mismatches in the "actual" and "assumed" dynamics, and the execution errors and measurement biases may then be varied to determine the effects of these parameters on the navigation and guidance process. All guidance and probe release event options defined for ERRAN are also available in SIMUL.

The documentation for STEAP II consists of three volumes: the Analytic, Programmers' and User's Manuals. Each of these documents is self-contained.

The *STEAP Analytic Manual* consists of two major divisions. The first section provides a unified treatment of the mathematical analysis of the STEAP II programs. The general problem description, formulation, and solution are given in a tutorial manner. The second section of this report supplies the detailed analysis of those subroutines of STEAP II dealing with technical tasks.

The *STEAP Programmers' Manual* provides the reader with the information he needs to modify the programs. Both the overall structure of the programs as well as the computational flow and analysis of the individual subroutines is described in this manual.

The *Users' Manual* contains the information necessary to operate the programs. The input and output quantities of the programs are described in detail. Example cases are also given and discussed.

CONTENTS

	<u>Page</u>
FOREWORD.	iii
1. INTRODUCTION	1
2. NOMENCLATURE	2
3. SUMMARY OF MODES	7
3.1 The Virtual Mass Propagator VMP	7
3.2 The Nominal Trajectory Targeter NOMNAL.	8
3.3 The Error Analysis Program ERRAN.	11
3.4 The Simulation Program SIMUL.	12-1
4. DESCRIPTION OF SUBROUTINES	13
4.1 Index of Subroutines.	13
4.2 VMP Subroutine Hierarchy.	13
4.3 NOMNAL Subroutine Hierarchy	14
4.4 ERRAN and SIMUL Hierarchy	14
5. COMMON VARIABLE DEFINITIONS	25
5.1 Common Variables by Blocks.	26
5.2 Common Variables in Alphabetical Order.	65
5.3 Large Array Definitions	94
6. INDIVIDUAL SUBROUTINE DOCUMENTATION.	97
BIBLIOGRAPHY.	615

1. INTRODUCTION

This Programmer's Manual is intended to supply the reader with sufficient information about the STEAP II programs to enable him to efficiently modify them. Both the overall structure of the programs and the computational flow of the individual subroutines are described in this manual.

This section describes the contents of the Programmer's Manual. Following this discussion the nomenclature used throughout the report is presented.

The third section of this manual describes the four basic components of STEAP II: the n-body trajectory propagation package, the nominal trajectory generator NOMNAL, the error analysis program ERRAN, and the simulation program SIMUL. The general purpose and capability of each of the programs is briefly summarized.

Chapter 4 of this volume examines the STEAP II programs from a more detailed viewpoint. The operational structure of each of the main components is described at the subroutine level. The individual subroutines are defined and cross-referenced according to the three main programs of STEAP II.

Chapter 5 contains the definitions of the variables appearing in common blocks throughout the programs. The variables are first listed according to the common blocks to which they belong. The programs requiring each of these common blocks are also noted. Following this all the common variables are listed in alphabetical order with their common blocks referenced. Tables detailing the definitions of large, frequently referenced common arrays are also provided.

Chapter 6 comprises the bulk of this volume. Each of the subroutines is documented in detail in alphabetical order. The purpose of the subroutine is supplied. Subroutines supported or required by the subroutine are listed. Arguments and interval variables of the subroutine are defined and usage of common variables is noted. The mathematical analysis upon which the subroutine is based is then discussed in full. Finally a flow chart of the computational flow of the subroutine is provided.

2. NOMENCLATURE

A. Arabic symbols

Symbol	Definition
a	Semi-major axis of conic
$B \cdot T$	Impact plane parameter
$B \cdot R$	Impact plane parameter
C_{xx_s}	Correlation between position/velocity state and solve-for parameters
C_{xu}	Correlation between position/velocity state and dynamic consider parameters
C_{xv}	Correlation between position/velocity state and measurement consider parameters
$C_{x_s u}$	Correlation between solve-for parameters and dynamic consider parameters
$C_{x_s v}$	Correlation between solve-for parameters and measurement consider parameters
e	Eccentricity of conic
E	Eccentric anomaly
f	True anomaly on conic
G	Observation matrix relating observables to dynamic consider parameter state
H	Observation matrix relating observables to position/velocity state
i	Inclination of conic (reference body equatorial)
J	Measurement residual covariance matrix
K	Kalman gain constant for position/velocity state
L	Observation matrix relating observables to measurement consider parameter state Mean longitude
M	Observation matrix relating observables to solve-for parameter state Mean anomaly

n_1	Dimension of solve-for parameter state
n_2	Dimension of dynamic consider parameter state
n_3	Dimension of measurement consider parameter state
p	Semilatus rectum of conic Probability density function
P	Position/velocity covariance matrix
\hat{P}	Unit vector to periapsis of conic
P_s	Solve-for parameter covariance matrix
Q	Dynamic noise covariance matrix
\tilde{Q}	Execution error matrix
\hat{Q}	Unit vector in plane of motion normal to P
r	Radius
r_{CA}	Radius of closest approach
r_{SI}	Radius of sphere of influence
R	Measurement noise covariance matrix
\underline{R}	Actual noise covariance matrix
\hat{R}	Unit vector normal to T in plane perpendicular to approach asymptote directed south ($R = S \times T$)
R_c	Target planet capture radius
S	Kalman gain constant for solve-for parameters
S_j	Velocity correction covariance matrix
\hat{S}	Approach or departure asymptote
t_{CA}	Time of closest approach to target body
t_{SI}	Time of intersection with sphere of influence of target body
Δt	Time interval
\hat{T}	Unit vector lying in ecliptic plane normal to \hat{S} . $(\hat{T} = \frac{\hat{S} \times \hat{K}}{ \hat{S} \times \hat{K} } \text{ where } \hat{K} \text{ is unit normal to ecliptic plane.})$
U_o	Dynamic consider parameter covariance matrix

v	Velocity
V_o	Measurement consider parameter covariance matrix
W_j	Target parameter covariance matrix
\hat{W}	Unit normal to orbital plane
X	Actual position/velocity state
\bar{X}	Targeted nominal position/velocity state
\tilde{X}	Most recent nominal position/velocity state

B. Greek Symbols

α	Auxiliary parameters
Γ_j	Guidance matrix
Γ	Flight path angle
δ	Declination of vector
Δv	Velocity increment
ϵ	Measurement residual Errors in target parameters
η_j	Variation matrix relating position/velocity variations to target conditions
θ_{xx_s}	State transition matrix partition associated with solve-for parameters
θ_{xu}	State transition matrix partition associated with dynamic consider parameters
θ	Longitude or right ascension
Λ_j	Projection of target condition covariance matrix W_j into the impact plane
μ	Gravitational constant of body
$\vec{\mu}$	Biased aimpoint
ν	Sampled measurement noise True anomaly

ρ	Magnitude of gaussian approximation for midcourse correction Correlation coefficient
σ	Standard deviation
Σ	Launch azimuth
\vec{T}	Target parameters
Φ	Targeting matrix State transition matrix for position/velocity state Latitude
χ	Sensitivity matrix
ψ_j	Matrix relating guidance corrections to target condition deviations
Ω	Longitude of ascending node
ω	Argument of periapsis
$\tilde{\omega}$	Longitude of periapsis

C. Subscripts

C	Control variable (P_C)
CA	Closest approach (r_{CA})
f	Final variable (t_f)
i	Initial variable (t_i)
j	Index of current guidance event (P_j)
k	Index of current measurement (P_k)
K	Knowledge variable (P_K)
s	Solve-for parameter (x_s)
SI	Sphere of influence (t_{SI})

D. Superscripts

A	Augmented variable (Φ^A)
T	Matrix transpose (Φ^T)
-1	Matrix inverse (Φ^{-1})

-	Variable immediately before instant (P_k^- or v^-)
+	Variable immediately after instant (P_k^+ or v^+)

E. Abbreviations

AU	Astronomical unit
CA	Closest approach to reference body
ERRAN	Error analysis program
FTA	Fixed time of arrival guidance policy
GHA	Greenwich hour angle
J.D.	Julian date (referenced either 0 ^{yr} or 1900 ^{yr})
km	Kilometers
M/C	Midcourse correction
NOMNAL	Nominal trajectory generation program
POI	Probability of impact
Q-L	Quasilinear filter event
S/C	Spacecraft
SF/C	Solve-for/consider
SIMUL	Simulation program
SOI	Sphere of influence
STM	State transition matrix
STEAP	<u>S</u> pace <u>T</u> rajectories <u>E</u> rror <u>A</u> nalys <u>i</u> s <u>P</u> rograms
VM	Virtual Mass
2VBP	Two variable B-plane guidance policy
3VBP	Three variable B-plane guidance policy

3. SUMMARY OF MODES

The Space Trajectory Error Analysis Programs (STEAP) consist of four subprograms or operational modes. The first mode, used as a subroutine by each of the other three programs, is the trajectory mode VMP by which an n-body trajectory (lunar or interplanetary) is propagated by the virtual mass technique. The second mode is the nominal trajectory generator or targeter (NOMNAL) by which a lunar or interplanetary trajectory meeting specified conditions is determined. The third mode is the error analysis program ERRAN in which the navigation and guidance characteristics of a nominal trajectory are analyzed by linearly propagating knowledge and control covariances along the trajectory. Finally the simulation mode SIMUL tests the mathematical models used in the navigation and guidance processes by modeling the tracking and correction of an "actual" trajectory. In this chapter a general description of each of these modes will be provided.

3.1 The Virtual Mass Propagator VMP

The dynamic model used by STEAP is supplied by the trajectory propagation package. The only external forces acting upon the spacecraft are assumed to be the gravitational forces of the celestial bodies considered in the integration. Both the spacecraft and the gravitational bodies are assumed to be point masses so neither spacecraft attitude nor planet asphericities are considered.

The celestial bodies to be in the integration are specified by the user and may include the sun, any of the nine planets, and the earth's moon. The motion of the planets about the sun and the moon about the earth are modeled by using mean ecliptic elements of date. If the user desires, each of the planets can be set in a fixed ellipse referenced to some epoch for speedier computation.

The coordinate system used in the integration is also specified by the user. The options available are either heliocentric ecliptic or barycentric ecliptic (nominally for lunar trajectories).

The actual scheme used in the propagation of the trajectory in the virtual mass or varicentric technique (see reference 15). No actual integration is performed by the trajectory mode; the key idea of the virtual mass technique is to build up an n-body trajectory by using a sequence of conic sections around a moving effective force center called the virtual mass. At each instantaneous moment along the trajectory, the combined effects of all the gravitational bodies can be viewed as resulting from a fictitious body of unique magnitude and position which is called the virtual mass. The computational pro-

cedure then assumes that over a small time interval, the motion of the spacecraft can be represented by a two-body conic section arc relative to this virtual mass. The complete trajectory is thus generated by a series of small arcs pieced together in steps while updating the position and magnitude of the effective force center. The main advantage of the virtual-mass technique is that the tedious numerical integration of the differential equations is avoided.

Another significant feature of the virtual-mass technique is its flexibility. By varying a simple parameter called the "accuracy level" related to the true anomaly increment of each step, trajectories ranging from a sequence of relatively few conic section arcs corresponding to a very approximate solution to those requiring a large number of arcs corresponding to highly accurate solutions may be generated.

3.2 The Nominal Trajectory Targeter NOMNAL

NOMNAL is responsible for the generation of a nominal trajectory for either lunar or interplanetary missions. The method of propagation in either case is the virtual-mass n-body integrator. The trajectory may be processed through a series of deterministic maneuvers including initial or injection targeting, subsequent retargeting, miniprobe targeting, and orbit insertion. A variety of target parameters are available for the targeting events. Both coplanar and nonplanar strategies are permitted in the orbit insertion maneuver.

If an initial state for the problem is known, this may be read in to start the trajectory. Otherwise NOMNAL generates its own zero iterate. In interplanetary missions this involves solving the Lambert time-of-flight equation for the massless planet trajectory that connects the desired initial and final positions in the specified time interval. Four options are available in describing these reference points.

Initial Point	Final Point
Launch Planet	Target planet
Launch Planet	Specified Point
Specified Point	Target Planet
Specified Point	Specified Point

If the initial point is referenced to the launch planet, a launch profile is consulted to generate a realistic set of injection conditions consistent with the heliocentric trajectory.

For lunar trajectories a slightly different procedure is used. The required data for the lunar zero iterate includes specification of the desired semimajor axis with respect to the moon, radius and time of closest approach to the moon, and inclination to the lunar equator. Then the zero iterate is generated by first targeting a patched conic trajectory and then a multiconic trajectory to the desired conditions.

A targeting event may be processed immediately after obtaining a zero iterate state or at any point along the nominal trajectory. At a targeting event the current velocity is refined to yield a trajectory satisfying target parameter constraints. The possible target parameter are:

- | | | | |
|--------|--------|----------------|---------|
| 1) TPS | 5) B•T | 9) SMA (Lunar) | 13) DCP |
| 2) TSI | 6) B•R | 10) XF | 14) RAP |
| 3) TCS | 7) RCA | 11) YF | 15) TPR |
| 4) TCA | 8) INC | 12) ZF | |

The targeting method to be used is specified by the user. Either a modified Newton-Raphson algorithm or a steepest descent/conjugate gradient technique may be used.

Orbit insertion events are also available in NOMNAL. At a specified time the spacecraft state relative to the target body is computed. The resulting conic trajectory relative to the target body is then compared with the desired orbit to determine the optimal time to make the insertion and the required correction. At the proper time the velocity correction is then implemented. Two strategies are permitted in the orbit insertion computation:

- 1) Coplanar - The desired semimajor axis, eccentricity, and periapsis shift of a coplanar orbit are specified;
- 2) Nonplanar - The desired plane of the postinsertion state is specified along with nominal values of the orbit elements.

The targeted correction, orbit insertion correction, or an externally supplied correction may be executed if desired. Two models are available for this implementation--a simple impulsive addition or a more complex multiple pulse model.

NOMNAL is also capable of targeting a set of three miniprobes to three specified target sites. Since achieving impacts at three specified points on the planet surface constitutes a six-degree-of-freedom constraint while only four miniprobe release controls are available, any targeting process can, at most, achieve a minimum-miss solution. NOMNAL uses as its miss-index a weighted sum of the squares of the distances between the respective actual and desired B-plane asymptote pierce points. The weighting factors, which are supplied by the user, indicate the relative importance of securing nearby impacts at the respective target sites. NOMNAL computes its weighted least-squares solution by a hybrid pseudo-inverse and steepest-descent algorithm. The initial control iterate is constructed by approximately targeting the first miniprobe to one of the target sites using a single Newton-Raphson step.

Finally the program integrates and records all segments of the nominal trajectory between guidance events from injection at the launch planet until the appropriate termination condition input by the user. For a conglomerate vehicle NOMNAL records the separate branches of the trajectory belonging to the main probe and miniprobes as well as to the bus.

3.3 The Error Analysis/Generalized Covariance Analysis Program ERRAN

The error analysis/generalized covariance analysis program ERRAN is a preflight mission analysis tool that is used to determine how selected error sources influence the orbit determination process for interplanetary or lunar missions.

In the error analysis mode, ERRAN provides three primary quantitative results: (1) knowledge covariance matrices, which provide a measure of how well the actual trajectory is known, (2) control covariance matrices, which when propagated forward to the target provide a measure of how well the nominal target conditions will be satisfied by the actual trajectory, and (3) statistical midcourse ΔV s, which provide a measure of the amount of fuel required for a successful mission.

In the generalized covariance analysis mode, ERRAN provides all of the above information plus corresponding "actual" statistical information. The three results discussed in the previous paragraph are all computed on the basis of statistical distributions assumed by the navigation filter to describe the significant error sources. In the generalized covariance analysis mode, "actual" knowledge covariances, control covariances, and statistical midcourse ΔV s are computed on the basis of statistical distributions that actually describe both error sources acknowledged by the navigation filter and the error sources ignored. The primary use of the generalized covariance analysis program is to study the sensitivity of filter performance to off-design conditions.

ERRAN allows for employing gain generators for user-specified linear recursive navigation filters. Two gain generators are currently available in ERRAN: (1) Kalman-Schmidt filter, and (2) equivalent recursive consider mode weighted-least-squares filter.

State transition matrices are required to propagate covariance matrices over an arbitrary interval of time. Three methods are available for computing the 6x6 position/velocity state transition matrix. The first two methods, which are analytical methods, are analytical patched conic and analytical virtual mass. The third method uses numerical differencing to compute the state transition matrix. To increase the accuracy of the analytical techniques over long intervals, a state transition matrix cascading option is also available. Augmented parameter state transition matrices are always computed using numerical differencing.

Up to 23 dynamic and measurement parameters may be solved-for or considered by the navigation filter. Parameters not acknowledged in design of the filter may be treated as ignore parameters when ERRAN is run in the generalized covariance analysis mode. The dynamic parameters include biases in the gravitational constants of the sun and the target planet and biases in the six orbital elements of the target planet. Measurement biases include biases in the locations of the three earth-based tracking stations, and biases in all measurements. Available measurement types are range, range-rate, star-planet angles, and apparent planet diameter measurements. Measurement noise for each measurement type is assumed to be constant.

The computational procedure in ERRAN is divided into basic cycle computations and event computations. Basic cycle computations are concerned with the propagation of covariances forward to a measurement time and processing the measurement. Events refer to a set of specialized computation, not directly concerned with measurement processing, that can be scheduled to occur at arbitrary times along the trajectory.

The four events available in ERRAN are eigenvector, prediction, guidance, and probe release. At an eigenvector event the position and velocity partitions of the knowledge covariance matrix are diagonalized to reveal geometric information about the size and orientation of the position and velocity navigation uncertainties. Associated hyperellipsoids are also computed. At a prediction event the most recent covariance matrix is propagated forward to some critical trajectory time to determine predicted navigation uncertainties in the absence of further measurements.

The guidance event is the most complex event and yields much useful information for preflight mission analysis. Several types of guidance events are available in ERRAN. At a midcourse guidance event the user can choose from three midcourse guidance policies. The midcourse guidance event can also be constrained to satisfy planetary quarantine requirements. At an orbital insertion guidance event the user can choose from two insertion policies. Options are also available for changing target conditions in midflight and retargeting the trajectory using nonlinear techniques, or for simply applying an externally supplied or precomputed ΔV at some arbitrary trajectory time. Two thrust models are available--impulse and impulse series. Execution error statistics are generated using an error model defined by a proportionality error, a resolution error, and two pointing angle errors. At a midcourse guidance event in ERRAN we also compute a statistical ΔV and the target condition covariance matrix both before and after the midcourse correction.

Probe release events provide the capability to study missions employing multiprobe spacecraft. The multiprobe spacecraft is modeled as (1) a primary vehicle, or bus, with thrusting capability, (2) a main probe, with no thrusting capability, and (3) three miniprobes located symmetrically on booms attached to the bus, with no thrusting capability, and released simultaneously with ΔV s provided by spinning the bus. Probe release events currently operate only in the error analysis mode of ERRAN. All measurement types and solve-for or consider parameters described previously are defined for all probes. Separate measurement schedules can be defined for the bus and the main probe. An additional measurement schedule can also be defined for all three miniprobes. Knowledge and control covariances are propagated for each probe in sequential fashion.

3.4 The Simulation Program SIMUL

The simulation program SIMUL is the most complex program in the STEAP set of programs. In SIMUL the validity of the navigation and guidance process is examined by simulating an actual mission. Spacecraft state estimates are generated in SIMUL, as well as knowledge covariance matrices. The results given by the error analysis program ERRAN become meaningful only when SIMUL shows that the estimated spacecraft trajectory converges, within reasonable bounds specified by the covariance matrix, to the simulated actual trajectory.

All state transition matrix, parameter augmentation, and measurement options described in section 3.3 are also available in SIMUL. As in ERRAN, the computational procedure in SIMUL is divided into basic cycle computations. The SIMUL basic cycle is concerned with the generation of state estimates and an actual trajectory, together with all quantities generated in the ERRAN error analysis basic cycle. Eigenvector and prediction events in SIMUL involve all computations performed in the corresponding ERRAN events. In addition, the SIMUL prediction event propagates state estimates forward to the time to which we are predicting.

All options available in the ERRAN guidance event (see section 3.3) are also available in the SIMUL guidance event. The treatment of the midcourse guidance event, however, is different in several respects. First, since an estimated spacecraft state is generated in SIMUL, an actual midcourse ΔV can be computed rather than a statistical ΔV as in ERRAN. Also, all linear midcourse ΔV s computed in SIMUL can be recomputed using nonlinear techniques.

Finally, since an actual trajectory is generated in SIMUL, actual target errors after the midcourse correction are also computed.

Probe release events are also available in SIMUL. In addition to propagating knowledge and control covariance matrices for each probe, SIMUL also generates state estimates for each probe.

4. DESCRIPTION OF SUBROUTINES

4.1 Index of Subroutines

The subroutines making up the STEAP programs are listed according to category in Table 4.1 following. The programs are divided into three general classes: the subroutines making up the virtual mass propagation package used by the three basic programs, the additional subroutines required by NOMNAL and then the additional subroutines used in ERRAN and SIMUL. In Table 4.2 the subroutines are listed again by category with a brief summary of their purpose. Thus Table 4.2 can be used to track down the subroutine in which a specific task is performed. The individual subroutines are then documented in detail in alphabetical order in Chapter 6.

4.2 VMP Subroutine Hierarchy

The executive program for the virtual mass n-body trajectory propagator is named VMP. The reader should investigate the detailed analysis and flow chart of VMP in the individual subroutine documentation in Chapter 6. The summaries of the subroutines of VMP are given in the first part of Table 4.2. The subroutines are conveniently divided into four general classes:

Conic	Subroutines based on conic approximations
Ephemeris	Subroutines used to compute the positions and velocities of the gravitational bodies at different times along the trajectory
Propagation	Subroutines used in the direct computation of the trajectory of the spacecraft moving under the influence of all the gravitational bodies
Input/Output	Subroutines processing either the input or output from the virtual mass trajectory propagation

The calling hierarchy of the virtual mass programs is given in Figure 4.1. All subroutines within a given block are at an identical level relative to the calling hierarchy unless they are enclosed by parentheses. Subroutines within parentheses are called by the preceding subroutine. Otherwise calls to subroutines are indicated by arrows. Thus all subroutines within blocks connected directly to VMP are called directly from VMP.

4.3 NOMNAL Subroutine Hierarchy

The first of the three independent programs of STEAP is the nominal trajectory targeter NOMNAL. The main program controlling the processing of the program goes under the same name. Reference is made to the complete documentation of NOMNAL in Chapter 6. The subroutine hierarchy of NOMNAL is provided in Figure 4.2. BLOCK DATA loads the planetary constants used by many of the subroutines; it is therefore available to all subroutines of NOMNAL. PRELIM reads the input data and calls ZERIT for the computation of a zero iterate if necessary. ZERIT in turn calls HELIO or LUNA for the actual computation of the interplanetary or lunar zero iterate respectively. NOMNAL calls TRJTRY for the propagation of the nominal trajectory between guidance maneuvers. TRJTRY of course calls the VMP package described in Figure 4.1. NOMNAL calls GIDANS for the actual processing of any guidance event. GIDANS calls VMP to initialize arrays for the other events. If a targeting event requires a zero iterate computation, ZERIT is called. Subroutine TARGET controls the targeting events; INSERS controls the insertion decision computations. NOMNAL calls EXCUTE for the execution of either of these two types of events. GIDANS calls MPPROP to execute a main-probe propagation event (i.e., to provide a time history of the main-probe trajectory). Finally GIDANS calls TPRTRG to carry out a miniprobe targeting event (i.e., to obtain the minimum-miss release controls, record the impact data, and print histories of the minimum-miss miniprobe trajectories.

4.4 ERRAN and SIMUL Hierarchy

The calling hierarchy of the subroutines used in ERRAN and SIMUL is shown in Figures 4.3 and 4.4, respectively. The similar structure of ERRAN and SIMUL is apparent from these two figures. All subroutines can be classified under one or more of the following categories: input, output, basic cycle (measurement processing), or events.

The calling hierarchy of the subroutines is indicated by the level of the subroutine in Figures 4.3 and 4.4. A given subroutine calls all subroutines that are directly connected to the subroutine and are located on the next lower level. For the purposes of clarity, the lowest level subroutine on a given branch is enclosed in parentheses. BLOCK DATA is shown connected to the main hierarchy with a dashed line to indicate that the constants stored in BLOCK DATA are available to all subroutines.

The complete documentation of all subroutines used in ERRAN and SIMUL is given in Chapter 5 of this document.

I.

A. Conic	B. Ephemeris	C. Propagation	D. Input/Output
1. CAREL	1. TIME	1. VMP	1. TRAPAR
2. ELCAR	2. BLOCK DATA	2. ESTMT	2. INPUTZ
3. IMPACT	3. ORB	3. VECTOR	3. PRINT
4. SOLPS	4. EPHEM	4. VMAS	4. SPACE
	5. CENTER		5. NEWPGE
	6. PECEQ		
	7. EULMX		
	8. SUBSOL		

II

A. Executive	B. Zero Iterate	C. Targeting		
1. EXCUTE	1. BATCON	7. LUNTAR	1. DESENT	
2. GIDANS	2. FLITE	8. MULCON	2. KTROL	
3. MPPROP	3. HELIO	9. MULTAR	3. TARGET	
4. NOMNAL	4. LAUNCH	10. SERIE	4. TARMAX	
5. PRELIM	5. LUNA	11. ZERIT	5. TAROPT	
6. TRJTRY	6. LUNCON			
D. Insertion	E. Pulsing Arc	F. Miniprobe Targeting		
1. COPINS	1. (BATCON)	1. SACOCS		
2. INSERS	2. PERHEL	2. TPPROP		
3. NONINS	3. PREPUL	3. TRRTRG		
	4. PULSEX			
G. Mathematical Functions and Operations	H. Conic	I. Ephemeris		
1. DINCOS	6. SCAD	1. CAREL	6. HYPT	1. EPHEM
2. DINSIN	7. SCAR	2. CONCAR	7. IMPACT	2. ORB
3. JACOB	8. THPSOM	3. DIMPCP	8. IMPCT	3. PECEQ
4. MATIN	9. USCALE	4. ELIPT	9. SPHIMP	4. SUBSOL
5. MATPY	10. UxV	5. HPOST	10. STIMP	

III. ERRAN and SIMUL Subroutines

A. Executive	C. Navigation	D. Event	E. Input/Output
1. ERRAN	1. NAVM	1. SETEVN	1. DATA
2. SIMUL	2. GNAVM	2. SETEVS	2. DATA1
	3. GAIN1	3. PRED	3. GDATA
B. Dynamic Model	4. GAIN2	4. PRESIM	4. SKEDM
	5. SCHED	5. BEPS	5. DATAS
1. NTM	6. TRAKM	6. BATCON	6. DATA1S
2. NTMS	7. TRAKS	7. ZRANS	7. CONURT
3. PSIM	8. TARPRL	8. ATANH	8. TRANS
4. NDTM	9. STAPRL	9. BPLANE	9. CORREL
5. PLND	10. MEMO	10. QUASI	10. STMPR
6. MUND	11. MENOS	11. GUIDM	11. SUB1
7. PCTM	12. BIAS	12. GUISIM	12. TITLE
8. CONCZ	13. RNUM	13. GUID	13. GPRINT
9. CASCAD	14. DYNO	14. GUIS	14. MOMENT
	15. DYNOS	15. VARADA	15. PRINT3
	16. GHA	16. VARSIM	16. PRNTS3
	17. JACOBI	17. PARTL	17. PRINT4
	18. HYEIS	18. BIAIM	18. PRNTS4
	19. EIGHY	19. POICOM	
	20. MEAN	20. QCOMP	
	21. SAVMAT	21. NONLIN	
		22. PULCOV	
		23. EXCUT	
		24. EXCUTS	
		25. PROBE	
		26. PROBES	
		27. MINIQ	
		28. NTRY	
		29. GENGID	
		30. ATCEGV	
		31. GQCOMP	

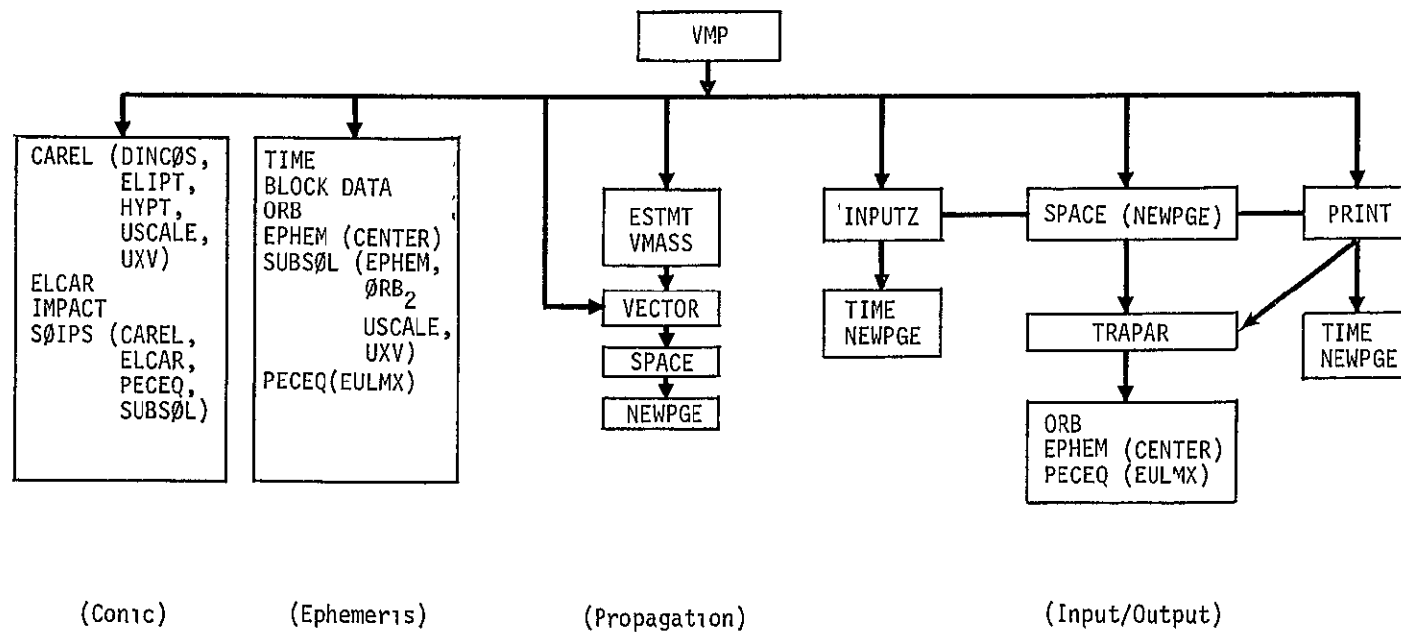


Figure 4.1 Subroutine Hierarchy of VMP

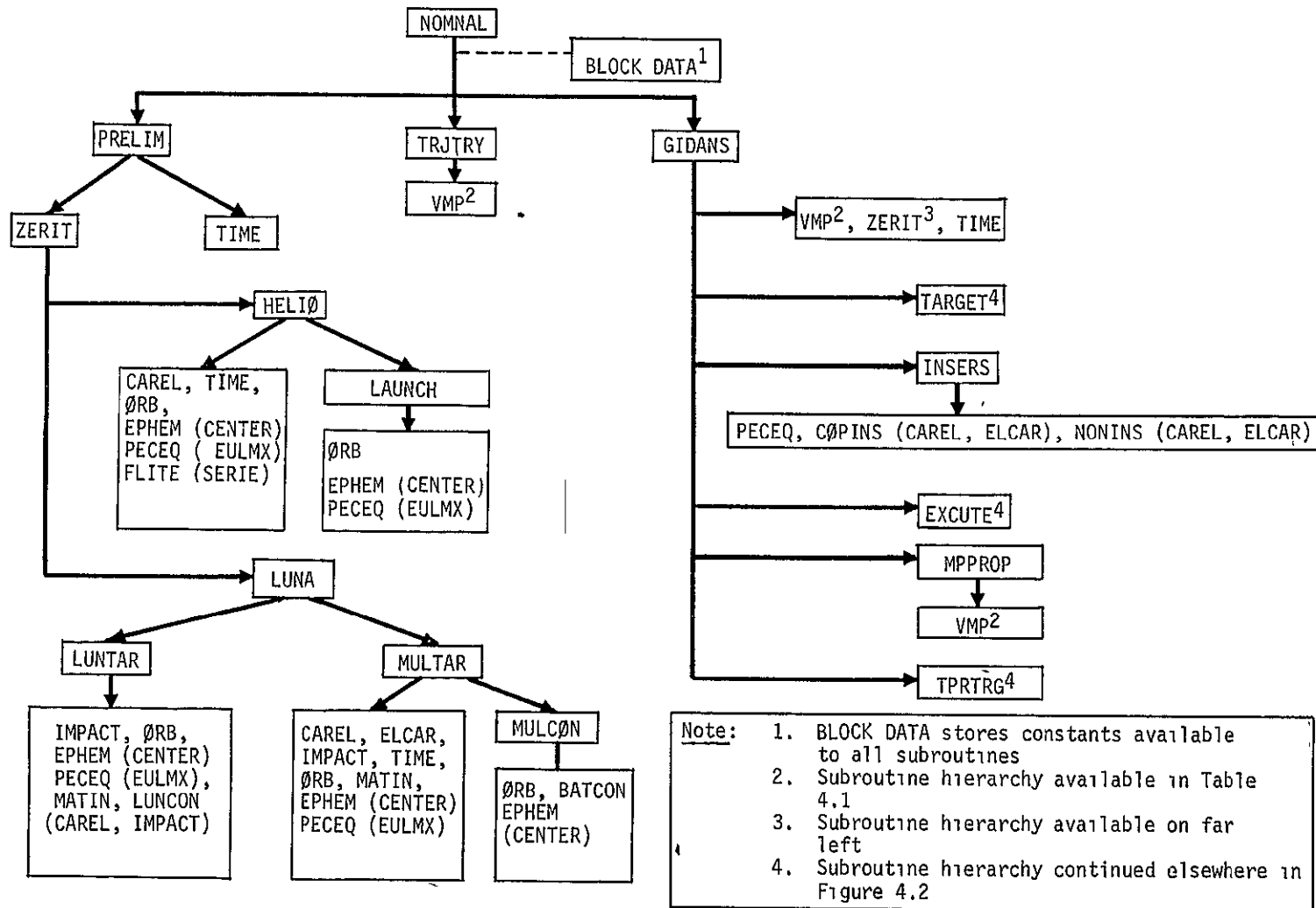
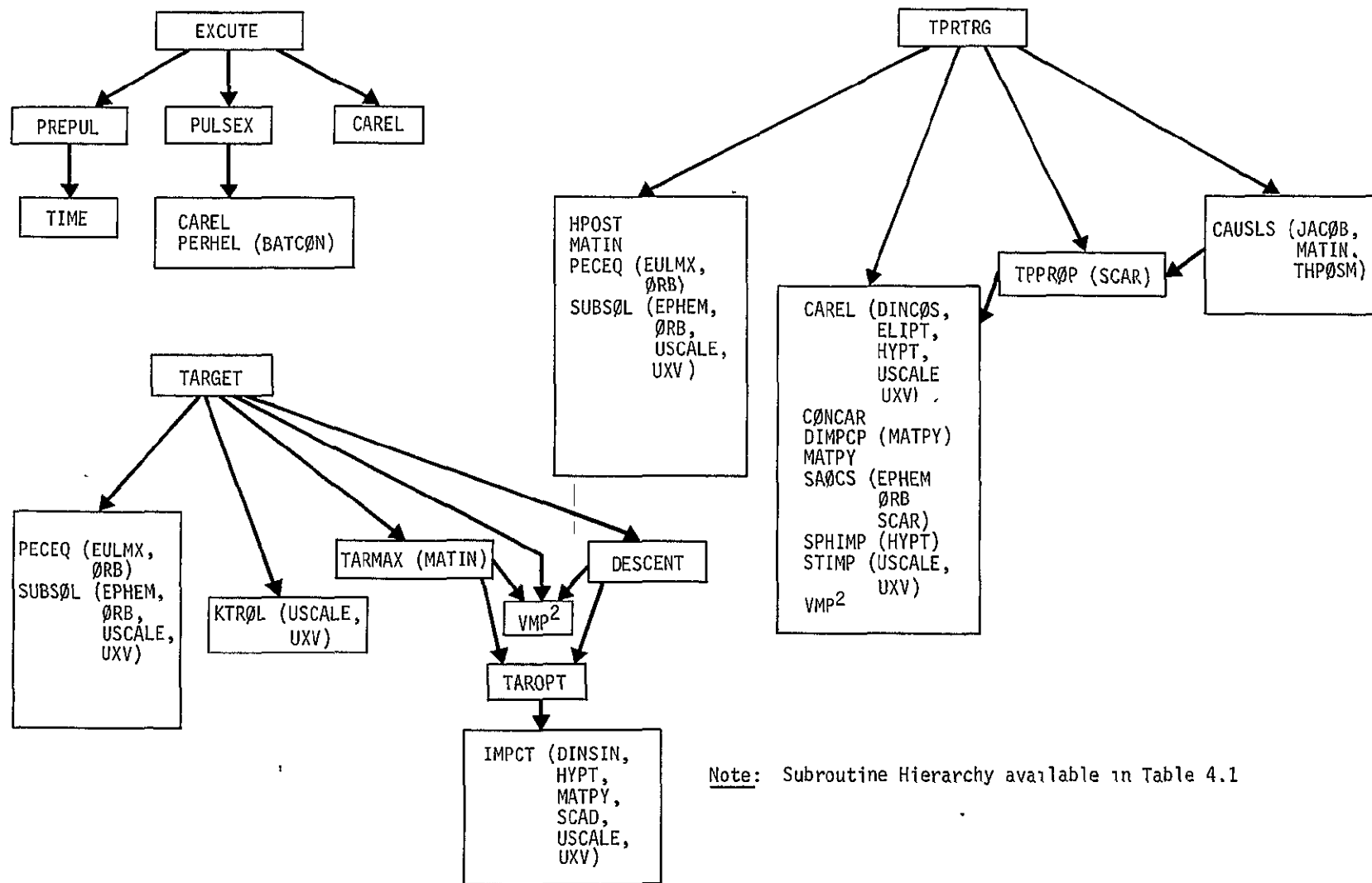


Figure 4.2a Subroutine Hierarchy of NOMNAL



Note: Subroutine Hierarchy available in Table 4.1

Figure 4.2b Subroutine Hierarchy of NOMNAL

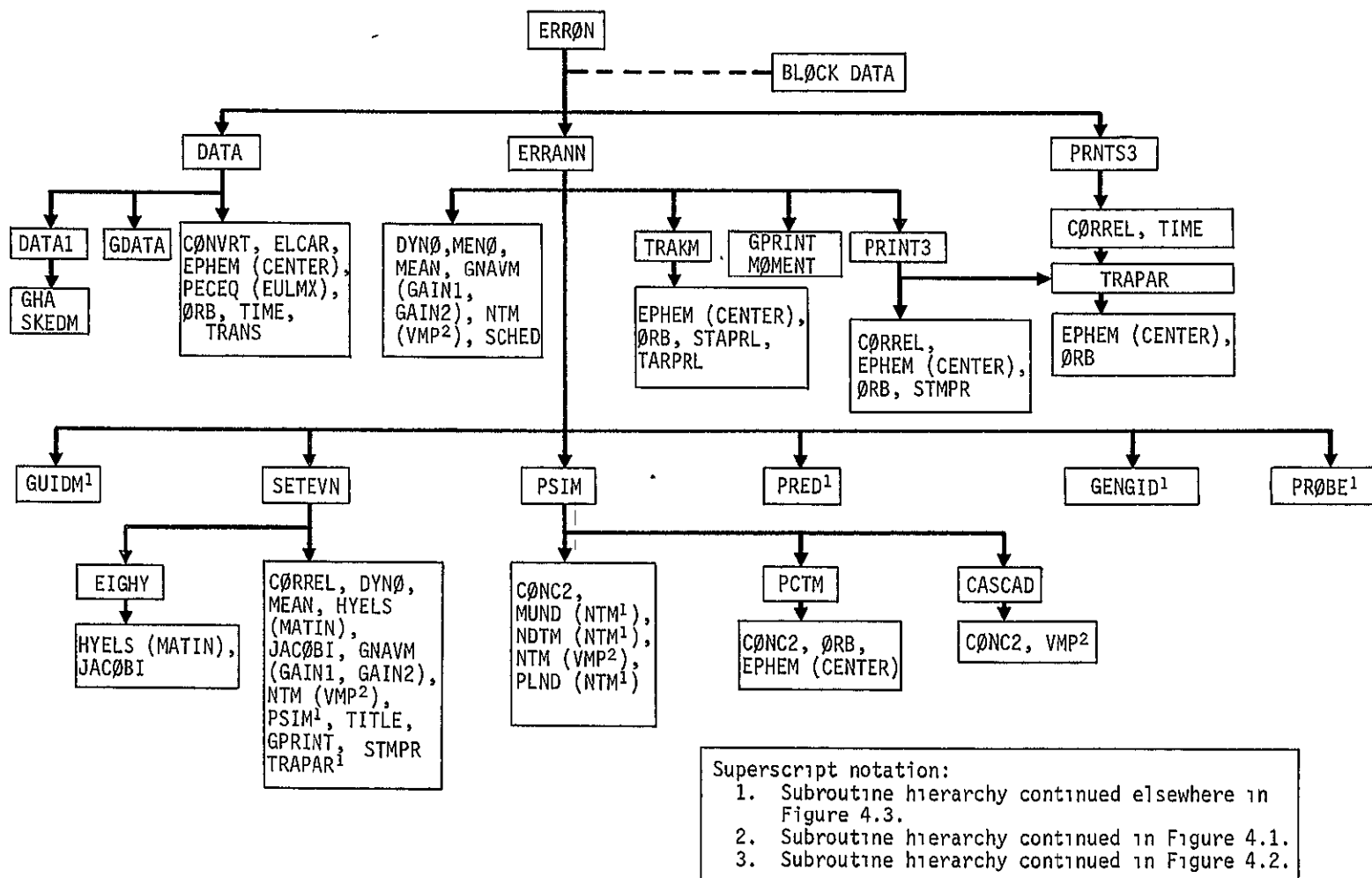


Figure 4.3a. Subroutine Hierarchy of ERRAN

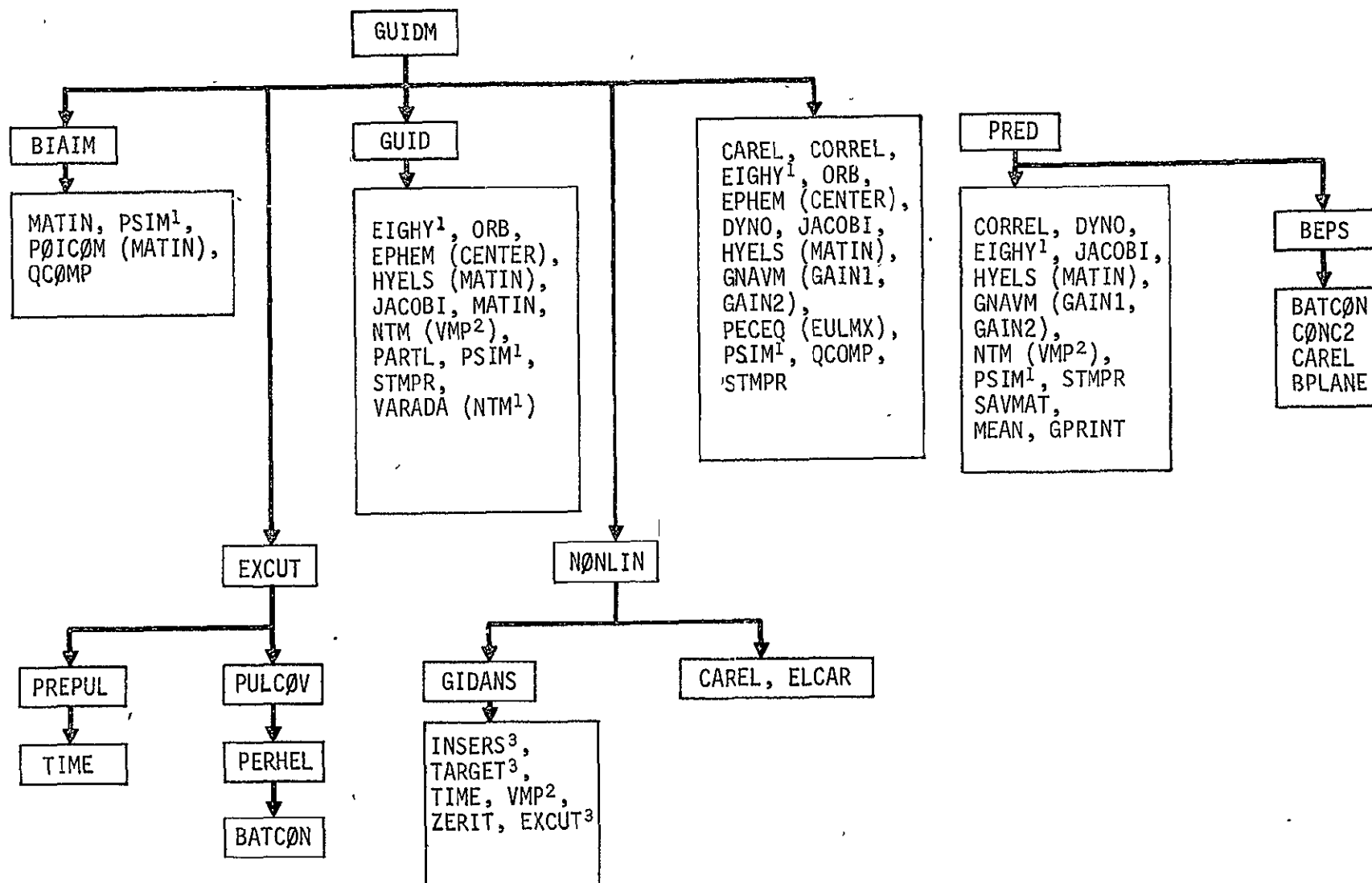


Figure 4.3b Subroutine Hierarchy of ERRAN

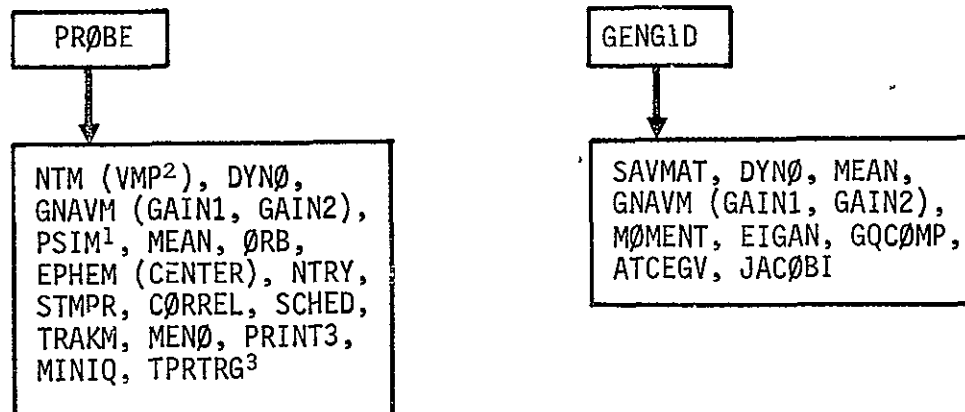


Figure 4.3c. Subroutine Hierarchy of ERRAN

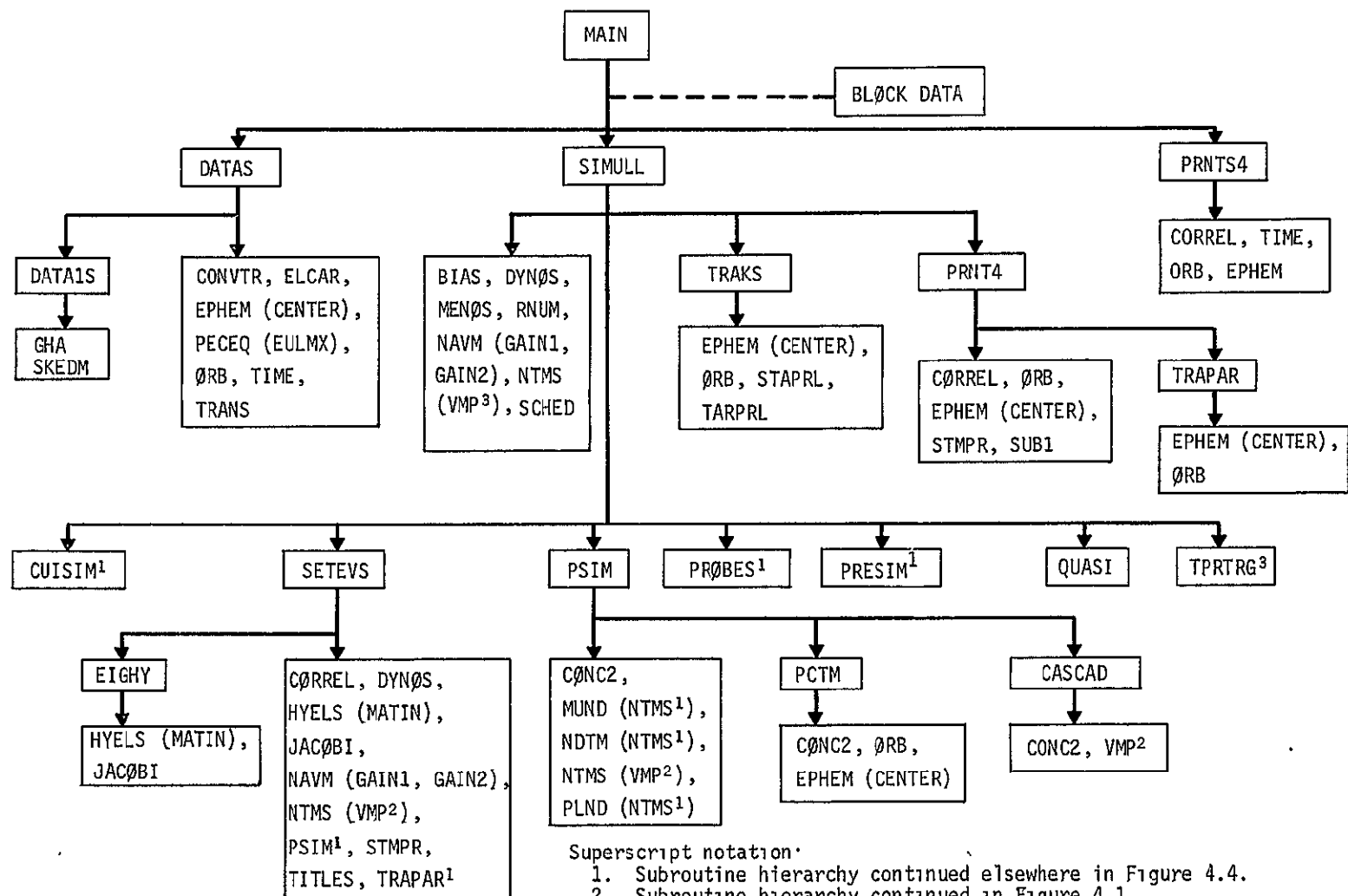


Figure 4.4a Subrouting Hierarchy of SIMUL

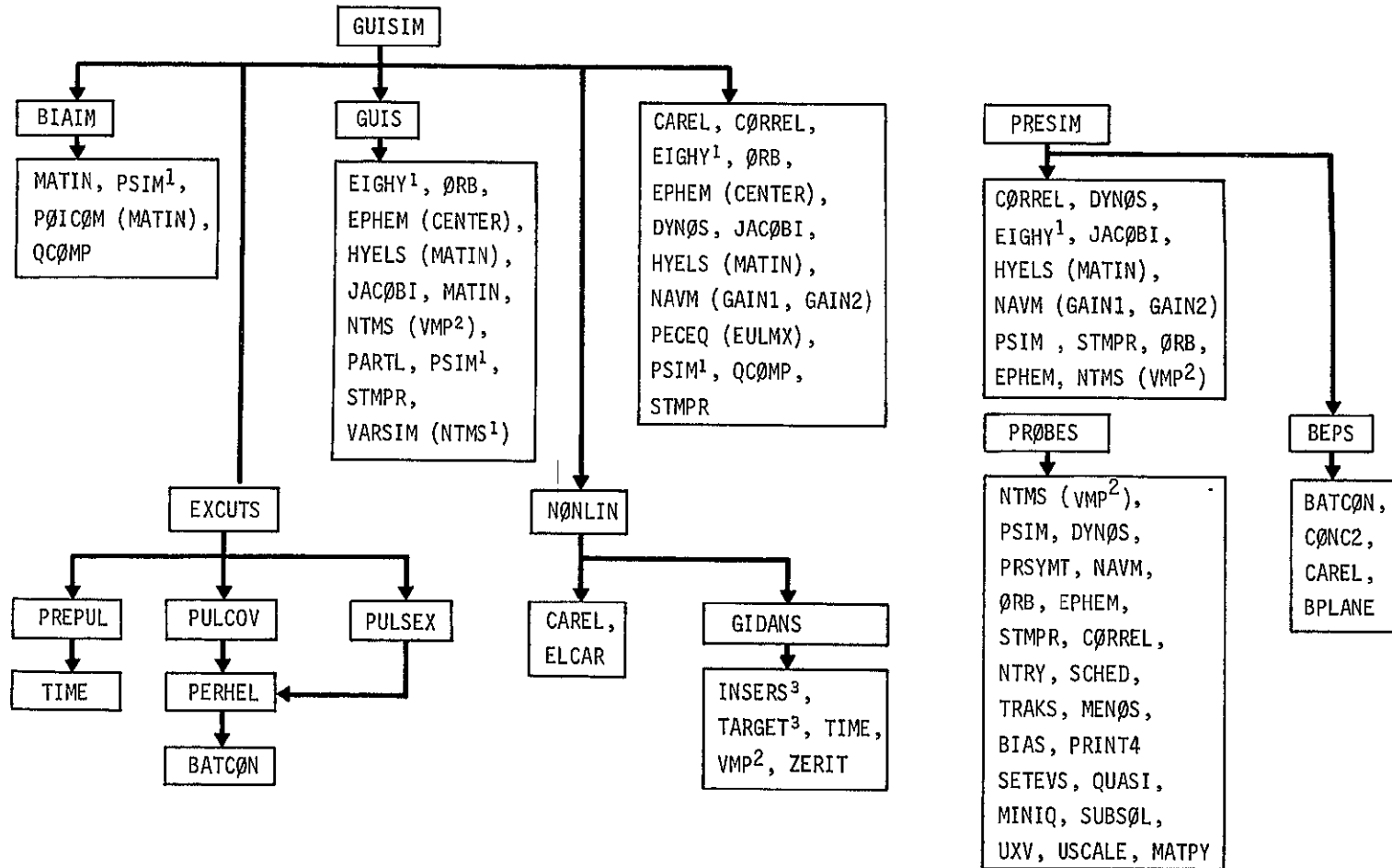


Figure 4.4b Subroutine Hierarchy of SIMUL (continued)

Table 4.2 STEAP II Subroutine Summaries

Subroutine	Function
I. Virtual-Mass Subroutines	
A. Conic	
1. CAREL	Convert a Cartesian state to conic elements
2. ELCAR	Convert conic elements to a Cartesian state
3. IMPACT	Compute the impact-plane parameters
4. SOIPS	Conically extrapolate from the nearest integration state to obtain impact data at the SOI and at the planet surface
B. Ephemeris	
1. BLOCK DATA	Set the ephemeris constants of the gravitational bodies
2. CENTER	Convert the states of bodies to barycentric coordinates
3. EPHEM	Compute the inertial state of a gravitational body at a given time
4. EULMX	Compute the rotational transformation matrix from the Euler angles
5. ORB	Compute the orbital elements of a gravitational body at a given time
6. PECEQ	Compute the transformation matrix from ecliptic to equatorial coordinates
7. SUBSOL	Compute the transformation matrix from ecliptic to subsolar coordinates
8. TIME	Convert Julian dates epoch 1900 to calendar dates or vice versa

Subroutine	Function
C. Propagation	
1. ESTMT	Determine final position and magnitude of the virtual mass on the current step
2. VECTOR	Compute the final position of the spacecraft on the current step
3. VMASS	Determine the virtual-mass data for the current step
4. VMP	Direct the virtual-mass trajectory propagation
D. Input/Output	
1. INPUTZ	Convert the input data into a form on which VMP can operate
2. NEWPGE	Print headings for each new page in VMP printout
3. PRINT	PRINT periodic trajectory-status data
4. SPACE	Space paper keeping tracking of paging
5. TRAPAR	Compute and record navigation parameter data
II. NOMNAL Subroutines	
A. Executive	
1. EXCUTE	Control the execution of a velocity-increment trajectory correction
2. GIDANS	Control the computation of a velocity-increment trajectory correction
3. MPPROP	Generate a time history of the main-probe trajectory

- | | |
|-----------------|--|
| 4. NOMNAL | Control the generation of the nominal trajectory (main program) |
| 5. PRELIM | Perform preliminary data processing for NOMNAL |
| 6. TRJTRY | Propagate the nominal trajectory to the next guidance event |
| B. Zero Iterate | |
| 1. BATCON | Propagate a conic trajectory by means of the universal conic functions |
| 2. FLITE | Obtain the solution to Lambert's time-of-flight equation |
| 3. HELIO | Compute the heliocentric phase of the interplanetary zero iterate |
| 4. LAUNCH | Compute the launch phase of the interplanetary zero iterate |
| 5. LUNA | Control lunar zero-iterate generation |
| 6. LUNCON | Generate a patched conic lunar trajectory |
| 7. LUNTAR | Control the patched conic targeting |
| 8. MULCON | Generate the lunar multiconic trajectory |
| 9. MULTAR | Control the lunar multiconic targeting |
| 10. SERIE | Compute the universal conic functions used in FLITE |
| 11. ZERIT | Control the computation of the zero iterate |
| C. Targeting | |
| 1. DESENT | Compute the interplanetary velocity targeting corrections using the descent scheme |

- | | |
|------------------------|---|
| 2. KTROL | Compute the heliocentric ecliptic velocity corrections given the launch-planetocentric velocity controls |
| 3. TARGET | Control the n-body targeting |
| 4. TARMAX | Compute the Newton-Raphson targeting matrix |
| 5. TAROPT | Set up the actual and auxiliary target parameter arrays |
| D. Insertion | |
| 1. COPINS | Compute the coplanar orbit insertion maneuver |
| 2. INSERS | Control the orbit insertion computation |
| 3. NONINS | Compute the nonplanar orbit insertion |
| E. Pulsing Arc | |
| 1. (BATCON) | Propagate a conic trajectory by means of the universal conic functions |
| 2. PERHEL | Propagate a perturbed heliocentric conic |
| 3. PREPUL | Perform the preliminary data processing for a multiple-pulse trajectory correction |
| 4. PULSEX | Execute pulsing arc |
| F. Miniprobe Targeting | |
| 1. SAOCS | Compute the sines and cosines of the spin-axis right ascension and declination given the spin-axis orientation mode |
| 2. TPPROP | Propagate the three miniprobe trajectories according to either a conic or virtual-mass model |

3. TPRTRG Control the manipprobe targeting procedure

G. Mathematical Functions and Operations

1. DINCOS Calculate in degrees the inverse cosine of a real number
2. DINSIN Calculate in degrees the inverse sine of a real number
3. JACOB Approximate by divided differences the Jacobian sensitivity matrix of a vector-valued function with respect to a vector variable
4. MATIN Invert a matrix of real-valued elements
5. MATPY Multiply two matrices of real-valued elements
6. SCAD Calculate both the sine and cosine of an angle given in degrees
7. SCAR Calculate both the sine and cosine of an angle given in radians
8. THPOSM Find the minimum of a function on a given interval by cubic interpolation
9. USCALE Scale the length of a three-vector to a specified real number
10. UXV Calculate the vector product of two three-vectors

H. Conic

1. CAREL Convert a Cartesian state to conic elements
2. CONCAR Convert a conic state in terms of r , θ , e , \underline{P} , \underline{Q} , and μ into a Cartesian state

- | | |
|--------------|---|
| 3. DIMPCP | Calculate the desired B-plane asymptote pierce-point coordinates given the right ascension and declination of a probe target site |
| 4. ELIPT | Calculate the time from periapsis on an ellipse given the true anomaly |
| 5. HPOST | Calculate the radius and true anomaly on a hyperbola given the time from periapsis |
| 6. HYPT | Calculate the time from periapsis on a hyperbola given the true anomaly |
| 7. IMPACT | Compute the impact-plane parameters |
| 8. IMPCT | For auxiliary targeting compute actual and desired B-plane asymptote pierce points as well as actual target values |
| 9. SPHIMP | Calculate the true anomaly and time from periapsis at which a conic approach trajectory pierces a planetocentric sphere of a given radius |
| 10. STIMP | Calculate the B-plane asymptote pierce-point coordinates of a conic trajectory given a state upon it |
| I. Ephemeris | |
| 1. EPHEM | Compute the inertial state of a gravitational body at a given time |
| 2. ORB | Compute the orbital elements of a gravitational body at a given time |
| 3. PECEQ | Compute the transformation matrix from ecliptic to equatorial coordinates |
| 4. SUBSOL | Compute the transformation matrix from ecliptic to subsolar coordinates |

III. ERRAN and SIMUL Subroutines

A. Executive

1. ERRAN Control error analysis program (main program)
2. SIMUL Control simulation program (main program)

B. Dynamic Model

1. NTM Control generation of trajectory data for ERRAN
2. NTMS Control generation of trajectory data for SIMUL
3. PSIM Control computation of state transition matrix (STM)
4. NDTM Compute unaugmented partition of STM by numerical differencing
5. PLND Compute STM partition associated with ephemeris biases
6. MUND Compute STM partition associated with gravitational constants
7. FCTM Compute unaugmented partition of STM by patched conic technique
8. CONC2 Compute unaugmented partition of STM by virtual-mass technique
9. CASCAD Compute unaugmented partition of STM by cascaded Darby matrizants

C. Navigation

1. NAVM Propagate covariance matrices between measurements and between events in SIMUL
2. GNAVM Propagate assumed and actual covariance matrices between measurements and between events in ERRAN

3. GAIN1	Compute the Kalman GAIN matrices
4. GAIN2	Compute the GAIN matrices for the equivalent recursive consider weighted-least-squares filter
5. SCHED	Select next measurement time from measurement schedule
6. TRAKM	Compute observation matrices
7. TRAKS	Compute observation matrices and actual measurements
8. TARPRL	Compute target planet position partials
9. STAPRL	Compute station location position and velocity partials
10. MENO	Compute assumed measurement noise covariance matrix
11. MENOS	Compute assumed and actual measurement noise covariance matrices
12. BIAS	Compute actual measurement bias
13. RNUM	Generate random numbers
14. DYNO	Compute dynamic noise covariance matrix
15. DYNOS	Compute dynamic noise covariance matrix and actual dynamic noise
16. GHA	Compute Greenwich hour angle
17. JACOBI	Compute eigenvalues and eigenvectors of a matrix
18. HYEELS	Compute hyperellipsoids
19. EIGHY	Control computation of eigenvalues, eigenvectors, and hyperellipsoids

20. MEAN Propagate and update means of actual state or parameter deviations and actual state of parameter estimation errors

21. SAVMAT Stores one vector in a second vector

D. Event

1. SETEVN Perform computations common to most events in ERRAN

2. SETEVS Perform computations common to most events in SIMUL

3. PRED Perform prediction event in ERRAN

4. PRESIM Perform prediction event in SIMUL

5. BEPS Compute B-Plane-Related covariances and state transition matrices

6. BATCON Compute trajectory data at time T given position and velocity at time 0

7. ZRANS Calculate transcendental functions used in the universal form of Kepler's equation

8. ATANH Find the angle Y whose TANH is X

9. BPLANE Compute B-plane parameters

10. QUASI Perform quasi-linear filtering event in SIMUL

11. GUIDM Perform guidance event in ERRAN

12. GUIDSIM Perform guidance event in SIMUL

13. GUID Compute guidance and variation matrices in ERRAN

14. GUI Compute guidance and variation matrices in SIMUL

15. VARADA Compute 3VBP variation matrix in ERRAN

16. VARSIM	Compute 3VBP variation matrix in SIMUL
17. PARTL	Compute partials of B•T, B•R, wrt state
18. BIAIM	Perform biased aimpoint guidance
19. POICOM	Compute probability of impact
20. QCOMP	Compute execution error covariance matrix
21. NONLIN	Control execution of nonlinear guidance events
22. PULCOV	Propagate covariance matrix across a series of pulses
23. EXCUT	Control execution of pulsing arc in ERRAN
24. EXCUTS	Control execution of pulsing arc in SIMUL
25. PROBE	Control execution of probe release events in ERRAN
26. PROBES	Control execution of probe release events in SIMUL
27. MINIQ	Compute execution error covariance matrix for miniprobe release
28. NTRY	Compute entry parameters, covariance, and communication angle
29. GENGID	Generalized covariance technique applied to guidance processes
30. ATCEGV	Compute eigenvalues and eigenvectors of actual target condition 2nd moment matrices
31. GQCOMP	Compute actual execution error statistics

E. Input/Output

- | | |
|------------|---|
| 1. DATA | Perform preliminary computations and read data in ERRAN |
| 2. DATA1 | Continuation of DATA |
| 3. GDATA | Initialized generalized covariance quantities |
| 4. SKEDM | Set up bus, main probe, and miniprobe measurement schedules |
| 5. DATAS | Perform preliminary computations and read data in SIMUL |
| 6. DATAS1 | Continuation of DATAS |
| 7. CONVRT | Convert JPL injection conditions to Cartesian components |
| 8. TRANS | Compute coordinate transformations |
| 9. CORREL | Compute and print correlation matrix partitions and standard deviations |
| 10. STMPR | Print \overline{STM} partitions |
| 11. SUB1 | Compute position and velocity magnitudes |
| 12. TITLE | Print titles |
| 13. GPRINT | Print actual estimation error statistics |
| 14. MOMENT | Convert 2nd moment matrices to correlation matrices and print them |
| 15. PRINT3 | Print basic cycle data in ERRAN |
| 16. PRNTS3 | Print ERRAN summary |
| 17. PKINT4 | Print basic cycle data in SIMUL |
| 18. PRNTS4 | Print SIMUL summary |

5. COMMON VARIABLE DEFINITIONS

THE BULK OF THE VARIABLES USED IN THE STEAP PROGRAMS ARE COMMON VARIABLES. THESE VARIABLES ARE DEFINED IN DETAIL IN THIS CHAPTER. THE FIRST SECTION LISTS THE COMMON BLOCKS IN ALPHABETICAL ORDER. THE PROGRAMS (NOMNAL, ERRAN, SIMUL) USING EACH COMMON BLOCK ARE NOTED. THE VARIABLES OF EACH COMMON BLOCK ARE DEFINED IN THE ORDER THAT THEY APPEAR IN THE COMMON BLOCK.

THE SECOND SECTION LISTS ALPHABETICALLY ALL VARIABLES APPEARING ANYWHERE IN COMMON. THE COMMON BLOCK TO WHICH THE VARIABLE BELONGS IS REFERENCED. THE DEFINITION OF THE VARIABLE IS THEN GIVEN.

THE THIRD SECTION SUPPLIES THE DEFINITIONS OF SEVERAL LARGE FREQUENTLY REFERENCED ARRAYS. THE ELEMENT APPEARING IN EACH COMPONENT OF EACH ARRAY IS NOTED.

5.1 COMMON VARIABLES BY BLOCKS

IN THIS SECTION COMMON BLOCKS APPEARING IN STEAP ARE LISTED IN ALPHABETICAL ORDER. VARIABLES WITHIN THESE BLOCKS ARE LISTED AND DEFINED IN THE ORDER THEY APPEAR IN THE PROGRAM.

/BAIN / MODE ERRAN, SIMUL

ATrans(6)	CLOSEST APPROACH STATE
TMPR(3)	MOST RECENT TARGET STATE
TNOMC(7)	NOMINAL CLOSEST APPROACH TARGET STATE, INCL. TIME
TNOMB(3)	NOMINAL B-PLANE TARGET STATE
PHI2(3,3)	INVERSE OF VARIATION MATRIX PARTITION
VINF	HYPERBOLIC EXCESS VELOCITY
TINJ	INJECTION TIME
PROBI	ALLOWABLE PROBABILITY OF IMPACT
ADA(3,6)	VARIATION MATRIX
T3(10)	ARRAY OF GUIDANCE EVENT TIMES
IBAG	NOT USED
IPQ	NOT USED
IGUID(5,10)	ARRAY OF GUIDANCE EVENT CODES
II	GUIDANCE EVENT COUNTER

/BLK / MODE: NOMNAL, ERRAN, SIMUL

T TRAJECTORY TIME IN DAYS

PMASS(11) GRAVITATIONAL CONSTANTS OF PLANETS IN
A.U.**3/DAY**2

CN(80) CONSTANTS USED TO CALCULATE THE ORBITAL
ELEMENTS OF THE FIRST FIVE PLANETS
(SEE LARGE ARRAY DEFINITIONS IN SECTION 5.3)

ST(50) CONSTANTS USED TO CALCULATE THE ORBITAL
ELEMENTS OF THE LAST FOUR PLANETS
(SEE LARGE ARRAY DEFINITIONS IN SECTION 5.3)

EMN(15) , THE CONSTANTS USED TO CALCULATE THE ORBITAL
ELEMENTS OF THE MOON
(SEE LARGE ARRAY DEFINITIONS IN SECTION 5.3)

SMJR(18) CONSTANTS USED TO CALCULATE THE SEMI-MAJOR
AXES OF THE PLANETS

RADIUS(11) THE RADIUS OF A GIVEN PLANET IN A.U.

RMASS(11) THE RELATIVE GRAVITATIONAL CONSTANT OF A
STATED PLANET WITH RESPECT TO THE SUN

ELMNT(80) CONTAINS THE ORBITAL ELEMENTS OF THE PLANETS
(SEE LARGE ARRAY DEFINITIONS IN SECTION 5.3)

SPHERE(11) THE SPHERES OF INFLUENCE OF THE PLANETS IN
A.U.

XP(6) THE POSITION AND VELOCITY OF A PLANET IN
INERTIAL ECLIPTIC COORDINATES

NO(11) AN ARRAY OF PLANET CODES BEING USED TO
GENERATE THE VIRTUAL MASS TRAJECTORY

/CNTRIC/ MODE: NOMNAL, ERRAN, SIMUL

IBARY REFERENCE COORDINATE SYSTEM CODE
 =0 HELIOCENTRIC COORDINATES
 =1 BARYCENTRIC COORDINATES

ICCOORD NON-FUNCTIONAL IN ERROR ANALYSIS MODE

INITAL NON-FUNCTIONAL IN ERROR ANALYSIS MODE

/COM / MODE: NOMNAL, ERRAN, SIMUL

V(16,7) AN ARRAY WHICH STORES PERTINENT VECTORS USED
 IN THE CALCULATION OF THE VIRTUAL MASS
 TRAJECTORY (SEE LARGE ARRAY DEFNS IN SECT 5.3)

F(44,4) CONTAINS THE POSITIONS AND VELOCITIES OF THE
 PLANETS AT A SPECIFIED TIME PLUS THE POSITIONS
 AND VELOCITIES OF THE SPACECRAFT RELATIVE TO
 THE PLANETS (SEE LARGE ARRAY DEFNS IN SECT 5.3)

PI THE VALUE OF THE MATHEMATICAL CONSTANT PI

RAD THE NUMBER OF DEGREES PER RADIAN

ITRAT IN INTERNAL CODE USED TO DETERMINE HOW MANY
 ITERATIONS HAVE BEEN ACCOMPLISHED IN THE
 VIRTUAL MASS PROCEDURE

KOUNT A CODE WHICH SPECIFIES WHETHER PRINT-OUT IS
 TO OCCUR AFTER THIS TIME INCREMENT

INCMNT NUMBER OF INCREMENTS USED

INCPR SPECIFIES AFTER HOW MANY TIME INCREMENTS
 PRINT-OUT IS TO OCCUR

INC DETERMINE WHETHER THE ABOVE OPTION IS TO BE
 USED

IPR A CODE WHICH DETERMINES IF PRINT-OUT IS TO
 OCCUR AFTER A SPECIFIED NUMBER OF DAYS

NBODYI	NUMBER OF BODIES CONSIDERED IN VIRTUAL MASS TRAJECTORY
NBODY	BASED ON ABOVE VALUE--EQUAL TO 4*NBODYI-3
IPRT(4)	SPECIFIES PRINT OPTIONS (IN STEAP TRAJECTORY THIS OPTION IS OMITTED. WHEN PRINT-OUT OCCURS ALL SECTIONS ARE AUTOMATICALLY PRINTED)
KL	PROBLEM NUMBER (NOMNAL ONLY)
IPG	PAGE NUMBER (NOMNAL ONLY)
LINCT	LINE COUNT (NOMNAL ONLY)
LINPGE	LINES PER PAGE (NOMNAL ONLY)

 /CONST / MODE: ERRAN, SIMUL

OMEGA	ROTATION RATE OF EARTH
EPS	OBLIQUITY OF EARTH
SAL(3)	ALTITUDES OF STATIONS
SLAT(3)	LATITUDES OF STATIONS
SLON(3)	LONGITUDES OF STATIONS
DNCN(3)	CONSTANTS FROM WHICH DYNAMIC NOISE IS COMPUTED
MNCN(12)	MEASUREMENT NOISE CONSTANTS
NST	NUMBER OF STATIONS TO BE USED (MAXIMUM 3)

/CONST2/ MODE: ERRAN, SIMUL

UST(3)	DIRECTION COSINE ARRAYS OF THREE REFERENCE STARS
VST(3)	DIIRECTION COSINE ARRAYS OF THREE REFERENCE STARS
WST(3)	DIRECTION COSINE ARRAYS OF THREE REFERENCE STARS
FOP	OFF-DIAGONAL ANNIHILATION VALUE FOR POSITION EIGENVALUES
FOV	OFF-DIAGONAL ANNIHILATION VALUE FOR VELOCITY EIGENVALUES

/CONST3/ MODE: ERRAN, SIMUL

DELAXS	TARGET PLANET SEMI-MAJOR AXIS FACTOR USED IN NUMERICAL DIFFERENCING
DELECC	TARGET PLANET ECCENTRICITY FACTOR USED IN NUMERICAL DIFFERENCING
DELICL	TARGET PLANET INCLINATION FACTOR USED IN NUMERICAL DIFFERENCING
DELNOD	TARGET PLANET LONGITUDE OF THE ASCENDING NODE FACTOR USED IN NUMERICAL DIFFERENCING
DELW	TARGET PLANET ARGUMENT OF PERIAPSIS FACTOR USED IN NUMERICAL DIFFERENCING
DELMA	TARGET PLANET MEAN ANOMALY FACTOR USED IN NUMERICAL DIFFERENCING
DELMUS	SUN GRAVITATIONAL CONSTANT FACTOR USED IN NUMERICAL DIFFERENCING
DELMUP	TARGET PLANET GRAVITATIONAL CONSTANT FACTOR USED IN NUMERICAL DIFFERENCING

/DPNUM / MODE: NOMNAL

ZERO	THE NUMBER ZERO (0) TO NINE SIGNIFICANT FIGURES
ONE	THE NUMBER ONE (1) TO NINE SIGNIFICANT FIGURES
TWO	THE NUMBER TWO (2) TO NINE SIGNIFICANT FIGURES
THREE	THE NUMBER THREE (3) TO NINE SIGNIFICANT FIGURES
FOUR	THE NUMBER FOUR (4) TO NINE SIGNIFICANT FIGURES
FIVE	THE NUMBER FIVE (5) TO NINE SIGNIFICANT FIGURES
EIGHT	THE NUMBER EIGHT (8) TO NINE SIGNIFICANT FIGURES
TEN	THE NUMBER TEN (10) TO NINE SIGNIFICANT FIGURES
NINETY	THE NUMBER NINETY (90) TO NINE SIGNIFICANT FIGURES
HALF	THE NUMBER ONE-HALF (1/2) TO NINE SIGNIFICANT FIGURES

 /DPNUM / MODE: ERRAN, SIMUL

ZERO	THE NUMBER ZERO (0) TO NINE SIGNIFICANT FIGURES
ONE	THE NUMBER ONE (1) TO NINE SIGNIFICANT FIGURES
TWO	THE NUMBER TWO (2) TO NINE SIGNIFICANT FIGURES
HALF	THE NUMBER ONE-HALF (1/2) TO NINE SIGNIFICANT FIGURES
THREE	THE NUMBER THREE (3) TO NINE SIGNIFICANT FIGURES
EM1	THE NUMBER 1.E-1 TO NINE SIGNIFICANT FIGURES
EM2	THE NUMBER 1.E-2 TO NINE SIGNIFICANT FIGURES
EM3	THE NUMBER 1.E-3 TO NINE SIGNIFICANT FIGURES
EM4	THE NUMBER 1.E-4 TO NINE SIGNIFICANT FIGURES
EM5	THE NUMBER 1.E-5 TO NINE SIGNIFICANT FIGURES
EM6	THE NUMBER 1.E-6 TO NINE SIGNIFICANT FIGURES
EM7	THE NUMBER 1.E-7 TO NINE SIGNIFICANT FIGURES
EM8	THE NUMBER 1.E-8 TO NINE SIGNIFICANT FIGURES
EM9	THE NUMBER 1.E-9 TO NINE SIGNIFICANT FIGURES
EM50	THE NUMBER 1.E-50 TO NINE SIGNIFICANT FIGURES
TWOPI	THE MATHEMATICAL CONSTANT 2.*PI
EM13	THE NUMBER 1.E-13 TO NINE SIGNIFICANT FIGURES

/EVENT / MODE: ERRAN, SIMUL

TEV(50)	TIMES OF EVENTS
TPT2(20)	PREDICTION TIMES
SIGRES	VARIANCE OF RESOLUTION ERROR
SIGPRO	VARIANCE OF PROPORTIONALITY ERROR
SIGALP	VARIANCE OF ERROR IN POINTING ANGLE 1
SIGBET	VARIANCE OF ERROR IN POINTING ANGLE 2
NEV	NUMBER OF EVENTS
IEVNT(50)	CODES OF EVENTS
IHYP1	HYPERELLIPSOID CODE USED TO DETERMINE IF K=1, K=3, OR BOTH
IEIG	CODE USED TO DECIDE IF BOTH POSITION AND VELOCITY EIGENVECTORS ARE REQUESTED
NPE	NUMBER OF PREDICTION EVENTS HAVING OCCURRED
NGE	NUMBER OF GUIDANCE EVENTS HAVING OCCURRED
ICDQ3(10)	ARRAY OF CODES WHICH DETERMINE WHICH EXECUTION POLICIES ARE TO BE USED IN GUIDANCE EVENTS
NEV1	TOTAL NUMBER OF EIGENVECTOR EVENTS
NEV2	TOTAL NUMBER OF PREDICTION EVENTS
NEV3	TOTAL NUMBER OF GUIDANCE EVENTS
NEV4	TOTAL NUMBER OF -COMCON- EVENTS
NQE	QUASI-LINEAR FILTERING EVENTS HAVING OCCURRED

IOPT7	ORBIT INSERTION VARIABLE. NON-FUNCTIONAL IN EXISTING PROGRAM
NEV8	ORBIT INSERTION VARIABLE. NON-FUNCTIONAL IN EXISTING PROGRAM
NEV9	ORBIT INSERTION VARIABLES. NON-FUNCTIONAL IN EXISTING PROGRAM
NEV10	ORBIT INSERTION VARIABLES. NON-FUNCTIONAL IN EXISTING PROGRAM
NEV11	ORBIT INSERTION VARIABLES. NON-FUNCTIONAL IN EXISTING PROGRAM

 /EXE / MODE ERRAN, SIMUL

XXIN(6)	STATE VECTOR TRANSFERRED TO EXCUT OR EXCUTS
DIPX	JULIAN DATE TRANSFERRED TO EXCUT OR EXCUTS
DELPX(3)	VELOCITY CORRECTION TO BE MODELED AS AN IMPULSE SERIES
QK(6,6)	EFFECTIVE EXECUTION COVARIANCE MATRIX
DUMMYQ(4)	ARRAY OF EXECUTION ERROR VARIANCES
INPX	IMPULSE SERIES CODE
EXMEAN(4)	ACTUAL IMPULSIVE EXECUTION ERROR MEANS

/GAINC / MODE: ERRAN

PMIN(6,6)	POSITION/VELOCITY COVARIANCE BEFORE MEASUREMENT USED TO COMPUTE WLS GAINS
PSMIN(12,12)	SOLVE-FOR PARAMETER COVARIANCE BEFORE MEASUREMENT USED TO COMPUTE WLS GAINS
CMIN(6,12)	CORRELATION BETWEEN POSITION/VELOCITY STATE AND SOLVE-FOR PARAMETERS BEFORE MEASUREMENT USED TO COMPUTE WLS GAINS
PPLU(6,6)	POSITION/VELOCITY COVARIANCE AFTER MEASUREMENT USED TO COMPUTE WLS GAINS
PSPLU(12,12)	SOLVE-FOR PARAMETER COVARIANCE AFTER MEASUREMENT USED TO COMPUTE WLS GAINS
CPLU(6,12)	CORRELATION BETWEEN POSITION/VELOCITY STATE AND SOLVE-FOR PARAMETERS AFTER MEASUREMENT USED TO COMPUTE WLS GAINS
RSAVE(6)	STATE AT TLAST
TLAST	TIME WHEN MEASUREMENT LAST PROCESSED

/GCA / MODE: ERRAN

XIG(24)	IGNORE PARAMETER LABELS
IAUGH(24)	IGNORE PARAMETER AUGMENTATION VECTOR
NDIM4	DIMENSION OF IGNORE PARAMETER STATE
IGEN	=0, PERFORM NO GENERALIZED COVARIANCE ANALYSIS =1, PERFORM GENERALIZED COVARIANCE ANALYSIS

/GENGD / MODE: ERRAN

EE(4) VECTOR WITH THE FOLLOWING ELEMENTS:
1 - ACTUAL MEAN OF PROPORTIONALITY ERROR
2 - ACTUAL MEAN OF RESOLUTION ERROR
3 - ACTUAL MEAN OF POINTING ANGLE ALPHA
ERROR
4 - ACTUAL MEAN OF POINTING ANGLE BETA
ERROR

EEE(4) VECTOR CONTAINING VARIANCES CORRESPONDING TO
THE -EE- VECTOR MEANS

/GENGD1/ MODE: ERRAN

GPG(6,6) ACTUAL POSITION/VELOCITY CONTROL SECOND MOMENT
MATRIX

GCXXSG(6,12) ACTUAL CONTROL SECOND MOMENT MATRIX OF POSITION/
VELOCITY STATE AND SOLVE-FOR PARAMETERS

G CXUG(6,8) ACTUAL CONTROL SECOND MOMENT MATRIX OF POSITION/
VELOCITY STATE AND DYNAMIC CONSIDER PARAMETERS

G CXVG(6,15) ACTUAL CONTROL SECOND MOMENT MATRIX OF POSITION/
VELOCITY STATE AND MEASUREMENT CONSIDER
PARAMETERS

G CXWG(6,12) ACTUAL CONTROL SECOND MOMENT MATRIX OF POSITION/
VELOCITY STATE AND IGNORE PARAMETERS

GPSG(12,12) ACTUAL SOLVE-FOR PARAMETER CONTROL SECOND MOMENT
MATRIX

G CXSUG(12,8) ACTUAL CONTROL SECOND MOMENT MATRIX OF SOLVE-FOR
PARAMETERS AND DYNAMIC CONSIDER PARAMETERS

G CXSVG(12,15) ACTUAL CONTROL SECOND MOMENT MATRIX OF SOLVE-FOR
PARAMETERS AND MEASUREMENT CONSIDER PARAMETERS

G CXSWG(12,12) ACTUAL CONTROL SECOND MOMENT MATRIX OF SOLVE-FOR
PARAMETERS AND IGNORE PARAMETERS

 /GENRL / MODE: ERRAN

GP(6,6)	ACTUAL POSITION/VELOCITY SECOND MOMENT MATRIX
GCXXS(6,12)	ACTUAL SECOND MOMENT MATRIX OF POSITION/VELOCITY STATE AND SOLVE-FOR PARAMETERS
GCXU(6,8)	ACTUAL SECOND MOMENT MATRIX OF POSITION/VELOCITY STATE AND DYNAMIC CONSIDER PARAMETERS
GCXV(6,15)	ACTUAL SECOND MOMENT MATRIX OF POSITION/VELOCITY STATE AND MEASUREMENT CONSIDER PARAMETERS
GPS(12,12)	ACTUAL SOLVE-FOR PARAMETER SECOND MOMENT MATRIX
GCXSU(12,8)	ACTUAL SECOND MOMENT MATRIX OF SOLVE-FOR AND DYNAMIC CONSIDER PARAMETERS
GCXSV(12,15)	ACTUAL SECOND MOMENT MATRIX OF SOLVE-FOR AND MEASUREMENT CONSIDER PARAMETERS
GCXSW(12,12)	ACTUAL SECOND MOMENT MATRIX OF SOLVE-FOR AND IGNORE PARAMETERS
JPR(4,4)	ACTUAL MEASUREMENT RESIDUAL SECOND MOMENT MATRIX
TXW(6,12)	STATE TRANSITION MATRIX-PARTITION ASSOCIATED WITH IGNORE PARAMETERS
AN(4,12)	OBSERVATION MATRIX ASSOCIATED WITH IGNORE PARAMETER STATE
GCUV(8,15)	ACTUAL SECOND MOMENT MATRIX OF DYNAMIC CONSIDER AND MEASUREMENT CONSIDER PARAMETERS
GCUW(8,12)	ACTUAL SECOND MOMENT MATRIX OF DYNAMIC CONSIDER AND IGNORE PARAMETERS

GCVW(15,12)	ACTUAL SECOND MOMENT MATRIX OF MEASUREMENT CONSIDER AND IGNORE PARAMETERS
GU(8,8)	ACTUAL DYNAMIC CONSIDER PARAMETER SECOND MOMENT MATRIX
GV(15,15)	ACTUAL MEASUREMENT CONSIDER PARAMETER SECOND MOMENT MATRIX
GW(12,12)	ACTUAL IGNORE PARAMETER SECOND MOMENT MATRIX
GDNCN(3)	CONSTANTS FROM WHICH ACTUAL DYNAMIC NOISE IS COMPUTED
GMNCN(12)	ACTUAL MEASUREMENT NOISE VARIANCES
EXI(6)	ACTUAL MEANS OF INITIAL POSITION/VELOCITY PARAMETER DEVIATIONS
EXSI(12)	ACTUAL MEANS OF INITIAL SOLVE-FOR PARAMETER DEVIATIONS
EU(8)	ACTUAL MEANS OF INITIAL DYNAMIC CONSIDER PARAMETER DEVIATIONS
EV(15)	ACTUAL MEANS OF INITIAL MEASUREMENT CONSIDER PARAMETER DEVIATIONS
EW(12)	ACTUAL MEANS OF INITIAL IGNORE PARAMETER DEVIATIONS
QPR(6,6)	ACTUAL DYNAMIC NOISE SECOND MOMENT MATRIX
RPR(4,4)	ACTUAL MEASUREMENT NOISE SECOND MOMENT MATRIX

GCXH(6,12)	ACTUAL SECOND MOMENT MATRIX OF POSITION/VELOCITY STATE AND IGNORE PARAMETERS
EXT(6)	ACTUAL MEANS OF UPDATED ESTIMATION ERRORS FOR POSITION/VELOCITY STATE
EXST(12)	ACTUAL MEANS OF UPDATED ESTIMATION ERRORS FOR SOLVE-FOR PARAMETERS
EXTP(6)	ACTUAL MEANS OF PROPAGATED ESTIMATION ERRORS FOR POSITION/VELOCITY STATE
EXSTP(12)	ACTUAL MEANS OF PROPAGATED ESTIMATION ERRORS FOR SOLVE-FOR PARAMETERS
GCXWP(6,12)	ACTUAL SECOND MOMENT MATRIX OF POSITION/VELOCITY STATE AND IGNORE PARAMETERS PRIOR TO PROCESSING A MEASUREMENT
GCXSWP(12,12)	ACTUAL SECOND MOMENT MATRIX OF SOLVE-FOR AND IGNORE PARAMETERS PRIOR TO PROCESSING A MEASUREMENT
EMRES(4)	ACTUAL MEASUREMENT RESIDUAL MEAN
IGDNF	ACTUAL DYNAMIC NOISE FLAG
IGMNF	ACTUAL MEASUREMENT NOISE FLAG

/GUI / MODE: ERRAN, SIMUL

PG(6,6)	POSITION/VELOCITY CONTROL COVARIANCE
CXXSG(6,12)	CONTROL CORRELATION BETWEEN POSITION/VELOCITY STATE AND SOLVE-FOR PARAMETERS
CXUG(6,8)	CONTROL CORRELATION BETWEEN POSITION/VELOCITY STATE AND DYNAMIC CONSIDER PARAMETERS
CXVG(6,15)	CONTROL CORRELATION BETWEEN POSITION/VELOCITY STATE AND MEASUREMENT CONSIDER PARAMETERS
PSG(12,12)	SOLVE-FOR PARAMETER CONTROL COVARIANCE
CXSUG(12,8)	CONTROL CORRELATION BETWEEN SOLVE-FOR PARAMETERS AND DYNAMIC CONSIDER PARAMETERS
CXSVG(12,15)	CONTROL CORRELATION BETWEEN SOLVE-FOR PARAMETERS AND MEASUREMENT CONSIDER PARAMETERS
XG(6)	POSITION/VELOCITY STATE AT MOST RECENT GUIDANCE EVENT
TG	TRAJECTORY TIME AT MOST RECENT GUIDANCE EVENT
EM(2,6)	VARIATION MATRIX RELATING POSITION/VELOCITY DEVIATIONS TO B.T AND B.R DEVIATIONS

/GXXL / MODE: ERRAN

GPL(34)	GUIDANCE EVENT POLICY LABEL ARRAY
---------	-----------------------------------

/IMPTAR/ MODE: NOMNAL

DCP ACTUAL TARGET VALUE OF PROBE SITE DECLINATION IN
 DEG RELATIVE TO PROBE-SPHERE FRAME (IT MAY DIFFER
 FROM INPUT VALUE IN DTAR IF INPUT TARGET SITE
 CAN NOT BE ACHEIVED)

DIN ACTUAL TARGET VALUE OF INCLINATION IN DEG AT
 CLOSEST APPROACH (IT MAY DIFFER FROM INPUT VALUE
 IN DTAR IF LATTER CAN NOT BE ACIEVED)

ECSS (3,3) TRANSFORMATION MATRIX FROM PLANETOCENTRIC
 ECLIPTIC TO SUBSOLAR COORDINATES

RAP ACTUAL TARGET VALUE OF PROBE SITE RIGHT-
 ASCENSION IN DEG RELATIVE TO PROBE-SPHERE FRAME
 (IT MAY DIFFER FROM INPUT VALUE IN DTAR IF
 INPUT TARGET SITE CAN NOT BE ACHIEVED)

/IMPTAR/ MODE ERRAN, SIMUL

ANG TARGET INCLINATION CONVERTED FROM INPUT FORMAT TO
 VALUE BETWEEN 0 AND 180 DEGREES AND SATISFYING
 APPROACH ASYMPTOTE CONSTRAINT

/JULY/ MODE: ERRANN

ALPHA ELEVATION ANGLE OF S/C FROM STATION

BETA AZIMUTH OF S/C FROM STATION

/LUNART/ MODE NOMNAL

DTAR(3)	TARGET VALUES OF SMA, B.T, AND B.R IN LUNAR TARGETING
PCON(3)	PERTURBATIONS IN CONTROLS (ALPHA,DELTA,THETA)
TTOL(3)	ALLOWABLE TOLERANCES IN SMA, B.T, B.R
BCON(3)	MAXIMUM STEP SIZES OF CONTROLS
RI(6)	GEOCENTRIC STATE OF S/C AT LUNAR SOI
RMQ(6)	GEOCENTRIC STATE OF CENTER OF MOON AT TSI IN EQUATORIAL COORDINATES
RSI(6)	SELENOCENTRIC STATE OF S/C AT LUNAR SOI
RME(6)	GEOCENTRIC STATE OF CENTER OF MOON IN ECLIPTIC COORDINATES AT TSI
DECLIN	DECLINATION OF APPROACH ASYMPTOTE WITH RESPECT TO LUNAR EQUATOR
OTAR(3)	DESIRED VALUES OF SMA, RCA, AND INC
TCA	J.D. OF TIME AT LUNAR CLOSEST APPROACH (DESIRED)
RCA	RADIUS OF CLOSEST APPROACH TO MOON (DESIRED)
SMA	SEMI-MAJOR AXIS OF LUNAR HYPERBOLA (DESIRED)
CAI	DESIRED CLOSEST APPROACH EQUATORIAL INCLINATION
RPE	RADIUS OF EARTH PARKING ORBIT
TSI	PROJECTED J.D. AT SOI INTERSECTION
EMU	GRAVITATIONAL CONSTANT OF EARTH (KM3/SEC2)

TSPH	RADIUS OF LUNAR SOI (KM)
EQLQ(3,3)	TRANSFORMATION MATRIX FROM EARTH-EQUATORIAL TO LUNAR EQUATORIAL COORDINATES
ITAG	FLAG SPECIFYING STAGE OF TARGETING =1 IN SMA TARGETING =0 IN SMA, INC, RCA TARGETING

 /MEAS / MODE: ERRAN, SIMUL

TMN(500)	TIMES OF MEASUREMENTS
MCODE(500)	ARRAY OF MEASUREMENT CODES
NMN	TOTAL NUMBER OF MEASUREMENTS
MCNTR	NUMBER OF MEASUREMENTS HAVING OCCURRED

 /MISC / MODE: ERRAN, SIMUL

ACC	ACCURACY FIGURE USED IN VIRTUAL MASS PROGRAM
FACP	POSITION FACTOR USED IN NUMERICAL DIFFERENCING
FACV	VELOCITY FACTOR USED IN NUMERICAL DIFFERENCING
BIA(12)	MEASUREMENT BIASES
IDNF	DYNAMIC NOISE FLAG
IC00R	STATE VECTOR CODE WHICH DETERMINES IN WHICH COORDINATE SYSTEM THE VECTOR IS READ IN
ITR	MODE FLAG
IMNF	MEASUREMENT NOISE FLAG
ISP2	SPHERE OF INFLUENCE FLAG

/MNPR / MODE: ERRAN, SIMUL

ALFA	RIGHT ASCENSION OF SPIN AXIS
DELT	DECLINATION OF SPIN AXIS
XPHI	ROLL RELEASE ANGLE (IN RADIANS)
ABW	MAGNITUDE OF SPIN AXIS
YYL	BOOM LENGTH
XEE(5)	PROBE RELEASE EXECUTION ERROR UNCERTAINTIES
QT(3,3)	MINI-PROBE EXECUTION ERROR COVARIANCE MATRIX
ADV(3)	ACTUAL MINI-PROBE RELEASE EXECUTION ERRORS
DW	ACTUAL SPIN-RATE EXECUTION ERROR
DA	ACTUAL SPIN-AXIS RIGHT ASCENSION EXECUTION ERROR
DL	ACTUAL BOOM LENGTH ERROR
DD	ACTUAL SPIN-AXIS DECLINATION EXECUTION ERROR
DP	ACTUAL RELEASE ANGLE EXECUTION ERROR

/NAME / MODE: ERRAN, SIMUL

EVNM(11)	EVENT NAME
MNNAME(12,3)	MEASUREMENT NAME
CMPNM(30)	COMPONENT NAME

/OVER / MODE: SIMUL

RF(6) FINAL TARGETED NOMINAL STATE VECTOR
RF1(6) FINAL MOST RECENT NOMINAL STATE VECTOR

/OVER1 / MODE: SIMUL

RI(6) INITIAL TARGETED NOMINAL STATE VECTOR
TEVN TIME OF CURRENT EVENT
RI1(6) INITIAL MOST RECENT NOMINAL STATE VECTOR
ICODE EVENT CODE
NAFC NON-FUNCTIONAL ADAPTIVE FILTER CODE
NR NUMBER OF ROWS IN THE OBSERVATION MATRIX

/OVERE / MODE: ERRAN

RF(6) FINAL TARGETED NOMINAL STATE

/OVERL / MODE ERRAN, SIMUL

DTIME TIME INTERVAL BETWEEN ORBITAL INSERTION DECISION
AND EXECUTION

/OVERR / MODE ERRAN

RI(6) STATE VECTOR AT EVENT TIME

TEVN EVENT TIME

NOGEN =1 CALL GENGID
 =0 DO NOT CALL GENGID

/OVERX / MODE ERRAN, SIMUL

IX NONLINEAR GUIDANCE CODE

JX GUIDANCE EVENT COUNTER

XIN(6) STATE VECTOR TRANSFERRED TO NONLIN

/OVERZ / MODE ERRAN, SIMUL

RF(6) FINAL TARGETED STATE VECTOR

IGP MIDCOURSE GUIDANCE POLICY CODE

GA(3,6) GUIDANCE MATRIX

/PBLK / MODE ERRAN, SIMUL

A(2,3) FTA IMPACT PLANE TRANSFORMATION MATRIX
XMUS(2) NOMINAL IMPACT PLANE TARGET STATE
EXEC(3,3) EXECUTION ERROR COVARIANCE MATRIX
CR CAPTURE RADIUS OF TARGET PLANET
POI PROBABILITY OF IMPACT
XLAM(2,2) PROJECTION OF TARGET CONDITION COVARIANCE MATRIX
 INTO THE IMPACT PLANE
XLAMI(2,2) INVERSE OF XLAM(2,2)
DVRB(3) VELOCITY CORRECTION REQUIRED TO REMOVE AIMPOINT
 BIAS
DVUP(3) UPDATE VELOCITY CORRECTION
PSTAR NOMINAL PROBABILITY DENSITY FUNCTION EVALUATED AT
 TARGET PLANET CENTER
DVN(3) COMMANDED VELOCITY CORRECTION TRANSFERRED TO
 BIAIM
DELV(3,10) ARRAY OF EXTERNALLY-SUPPLIED VELOCITY CHANGES
IIGP MIDCOURSE GUIDANCE POLICY CODE
IEND =2 FOR 2 VARIABLE B-PLANE
 =3 FOR 3 VARIABLE B-PLANE
IBIAS BIASED AIMPOINT GUIDANCE EVENT FLAG
 = 0 AIMPOINT NOT BIASED
 = 1 AIMPOINT BIASED
IDENS PROBABILITY DENSITY FUNCTION CODE. NON-FUNCTIONAL

 /PRBE / MODE: ERRAN, SIMUL

PMN(12)	MEASUREMENT NOISE VARIANCES USED FOR MAIN PROBE
T6	MAIN PROBE RELEASE EVENT TIME
T7	MINI-PROBE RELEASE EVENT TIME
RPS	RADIUS OF PROBE SPHERE
SMN(12)	MEASUREMENT NOISE VARIANCES USED FOR MINI-PROBES
TIMPCT	APPROXIMATE TRAJECTORY TIME OF IMPACT FROM NOMNAL PROGRAM
TMN1(100)	MAIN PROBE MEASUREMENT SCHEDULE TIMES
TMN2(100)	MINI-PROBE MEASUREMENT SCHEDULE TIMES
IUTC	=1 TARGET CONTROLS DATA SUPPLIED BY USER =0 COMPUTE TARGET CONTROLS
NMNP(2)	NUMBER OF MEASUREMENTS TO BE PROCESSED (1)= MAIN PROBE (2)= MINI-PROBE
MCODE1(100)	ARRAY OF MAIN PROBE MEASUREMENT CODES
MCODE2(100)	ARRAY OF MINI-PROBE MEASUREMENT CODES
NENT1	NUMBER OF CARDS IN MEASUREMENT INPUT FOR MAIN PRB
NENT2	NUMBER OF CARDS IN MEASUREMENT INPUT FOR MINI-PRB
MCNTRP	MEASUREMENT COUNTER FOR PROBE RELEASE

/PROBD / MODE: NOMNAL

RPSP RADIUS OF PROBE IMPACT SPHERE IN KM
IPCSP FLAG INDICATING TYPE OF PLANETOCENTRIC
 COORDINATE SYSTEM FOR SPECIFYING PROBE IMPACT
 SITES
 =0 EQUATORIAL
 =1 SUBSOLAR ORBIT-PLANE

/PRT / MODE NOMNAL

MONTH(12) NAMES OF MONTHS
PLANET(11) NAMES OF GRAVITATIONAL BODIES

/PRT / MODE: ERRAN, SIMUL

PLANET(11) NAMES OF PLANETS

 /PULS / MODE NOMNAL

PULMAG	THRUST MAGNITUDE OF PULSING ENGINE
PULMAS	NOMINAL MASS OF SPACECRAFT DURING PULSING ARC
DUR	DURATION OF SINGLE PULSE
DTI	TIME INTERVAL (DAYS) BETWEEN SUCCESSIVE PULSES
DVI(3)	VELOCITY INCREMENT ADDED ON TYPICAL PULSE
DVF(3)	VELOCITY INCREMENT ADDED ON FINAL PULSE
PULT	TOTAL TIME INTERVAL OF PULSING ARC
RK(2,3)	POSITION VECTORS OF LAUNCH AND TARGET PLANETS AT IMPULSIVE TIME (MIDPOINT OF PULSING ARC)
VK(2,3)	VELOCITY VECTORS OF LAUNCH AND TARGET PLANETS AT IMPULSIVE TIME (MIDPOINT OF PULSING ARC)
FS(2,5)	F-SERIES COEFFICIENTS OF LAUNCH AND TARGET BODIES
GS(2,4)	G-SERIES COEFFICIENTS OF LAUNCH AND TARGET BODIES
GG(3)	GRAVITATIONAL CONSTANTS OF SUN, LAUNCH, AND TAR- GET BODIES
PSIGS	PULSING ARC ERROR MODEL RESOLUTION VARIANCE
PSIGK	PULSING ARC ERROR MODEL PROPORTION VARIANCE
PSIGA	PULSING ARC ERROR MODEL POINTING ANGLE A VARIANCE
PSIGB	PULSING ARC ERROR MODEL POINTING ANGLE B VARIANCE
NPUL	NUMBER OF PULSES IN PULSING ARC

/SAVVAL/ MODE ERRAN, SIMUL

XBDT	ORIGINAL VALUE OF B.T IN NONLINEAR GUIDANCE
XBDR	ORIGINAL VALUE OF B.R IN NONLINEAR GUIDANCE
XDSI	ORIGINAL VALUE OF TSI IN NONLINEAR GUIDANCE
XRSI(3)	ORIGINAL VALUE OF RSI IN NONLINEAR GUIDANCE
XVSI(3)	ORIGINAL VALUE OF VSI IN NONLINEAR GUIDANCE
XRC(6)	ORIGINAL VALUE OF RC IN NONLINEAR GUIDANCE
XDC	ORIGINAL VALUE OF DC IN NONLINEAR GUIDANCE

/SIMCNT/ MODE: SIMUL

DMUSB	BIAS IN GRAVITATIONAL CONSTANT OF SUN
DMUPB	BIAS IN GRAVITATIONAL CONSTANT OF TARGET PLANET
DAB	BIAS IN SEMI-MAJOR AXIS OF TARGET PLANET
DEB	BIAS IN ECCENTRICITY OF TARGET PLANET
DIB	BIAS IN INCLINATION OF TARGET PLANET
DN0B	BIAS IN LONGITUDE OF ASCENDING NODE
DWB	BIAS IN ARGUMENT OF PERIAPSIS
DMAB	BIAS IN MEAN ANOMALY
TTIM1	FIRST TIME USED FOR UNMODELLED ACCELERATION
TTIM2	SECOND TIME USED FOR UNMODELLED ACCELERATION

UNMAC(3,3)	UNMODELLED ACCELERATION
SLB(9)	BIASES IN STATION LOCATION CONSTANTS
AVARM(12)	VARIANCE OF ACTUAL MEASUREMENT NOISE
ARES(20)	ACTUAL RESOLUTION ERROR
APRO(20)	ACTUAL PROPORTIONALITY ERROR
AALP(20)	ACTUAL ERROR IN POINTING ANGLE 1
ABET(20)	ACTUAL ERROR IN POINTING ANGLE 2
IAMNF	ACTUAL MEASUREMENT NOISE FLAG

/SIM1 / MODE: SIMUL

XI1(6)	INITIAL STATE VECTOR OF MOST RECENT NOMINAL TRAJECTORY
XF1(6)	FINAL STATE VECTOR OF MOST RECENT NOMINAL TRAJECTORY
ADEVX(6)	ACTUAL DEVIATION IN THE STATE VECTOR
ADEVXS(24)	ACTUAL DEVIATION IN SOLVE-FOR PARAMETERS
EDEVX(6)	ESTIMATED DEVIATION IN THE STATE VECTOR
EDEVXS(24)	ESTIMATED DEVIATION IN SOLVE-FOR PARAMETERS
W(6)	ACTUAL DYNAMIC NOISE
ZI(6)	INITIAL ACTUAL STATE VECTOR
ZF(6)	FINAL ACTUAL STATE VECTOR AFTER ADDING EFFECT OF UNMODELED ACCELERATION
ANOS(4)	ACTUAL WHITE NOISE
RES(4)	RESIDUAL
EY(4)	ESTIMATED MEASUREMENT
AY(4)	ACTUAL MEASUREMENT
AR(4,4)	ACTUAL MEASUREMENT NOISE
ADEVXB(6)	ACTUAL DEVIATION IN STATE VECTOR AT BEGINNING OF TRAJECTORY
ADEVSB(24)	ACTUAL DEVIATION IN SOLVE-FOR PARAMETERS AT BEGINNING OF TRAJECTORY
AYMEY(4)	ACTUAL MEASUREMENT MINUS ESTIMATED MEASUREMENT
EDEVXM(6)	ESTIMATED DEVIATION IN THE STATE VECTOR (FOR ADAPTIVE FILTERING)
EDEVSM(24)	ESTIMATED DEVIATION IN SOLVE-FOR PARAMETERS (FOR ADAPTIVE FILTERING)

/SIM2 / MODE: SIMUL

NB1(11) ARRAY OF PLANET CODES IN ACTUAL TRAJECTORY
ACC1 ACCURACY USED IN ACTUAL TRAJECTORY
NBOD1 NUMBER OF BODIES IN ACTUAL TRAJECTORY

/SOIVMP/ MODE: NOMNAL, ERRAN, SIMUL

DCIMP DECLINATION OF VEHICLE IN DEG RELATIVE TO
 PLANETOCENTRIC PROBE-SPHERE FRAME AT IMPACT
D JULIAN DATE OF IMPACT WITH SPHERE OF INTEREST
 ON OSCULATING PLANETOCENTRIC CONIC
DEPOC JULIAN DATE EPOCH 1900 OF IMPACT WITH SPHERE OF
 INTEREST ON OSCULATING PLANETOCENTRIC CONIC
IP INDEX IDENTIFYING IMPACTED PLANET
RAIMP RIGHT ASCENSION OF VEHICLE IN DEG RELATIVE TO
 PLANETOCENTRIC PROBE-SPHERE FRAME AT IMPACT

/STM / MODE: ERRAN, SIMUL

P(6,6)	POSITION/VELOCITY COVARIANCE
CXXS(6,12)	CORRELATION BETWEEN POSITION/VELOCITY STATE AND SOLVE-FOR PARAMETERS
CXU(6,8)	CORRELATION BETWEEN POSITION/VELOCITY STATE AND DYNAMIC CONSIDER PARAMETERS
CXV(6,15)	CORRELATION BETWEEN POSITION/VELOCITY STATE AND MEASUREMENT CONSIDER PARAMETERS
PS(12,12)	SOLVE-FOR PARAMETER COVARIANCE
CXSU(12,8)	CORRELATION BETWEEN SOLVE-FOR PARAMETERS AND DYNAMIC CONSIDER PARAMETERS
CXSV(12,15)	CORRELATION BETWEEN SOLVE-FOR PARAMETERS AND MEASUREMENT CONSIDER PARAMETERS
U0(8,8)	DYNAMIC CONSIDER PARAMETER COVARIANCE MATRIX

VO(15,15) MEASUREMENT CONSIDER PARAMETER COVARIANCE
MATRIX

NOTE IF THE ENTIRE COVARIANCE MATRIX WERE ASSEMBLED FROM THE
GIVEN PARTITIONS THE RESULTANT MATRIX WOULD BE P(41,41)
WITH THE SYMMETRIC STRUCTURE-

	P(6,6)	CXXS(6,12),	CXU(6,8)	CXV(6,15)

P(41,41) =		PS(12,12)	CXSU(12,8)	CXSV(12,15)

			UO(8,8)	CUV(8,15)

				VO(15,15)

PHI(6,6) POSITION/VELOCITY STATE TRANSITION MATRIX

TXXS(6,12) STATE TRANSITION MATRIX PARTITION ASSOCIATED WITH
SOLVE-FOR PARAMETERS

TXU(6,8) STATE TRANSITION MATRIX PARTITION ASSOCIATED WITH
DYNAMIC CONSIDER PARAMETERS

Q(6,6) DYNAMIC NOISE COVARIANCE MATRIX

R(4,4) MEASUREMENT NOISE COVARIANCE MATRIX

AK(6,4) KALMAN GAIN CONSTANT FOR POSITION/VELOCITY
STATE

S(12,4) KALMAN GAIN CONSTANT FOR SOLVE-FOR
PARAMETERS

H(4,6) OBSERVATION MATRIX RELATING OBSERVABLES TO
POSITION/VELOCITY STATE

AM(4,12) OBSERVATION MATRIX RELATING OBSERVABLES
TO SOLVE-FOR PARAMETER STATE

G(4,8) OBSERVATION MATRIX RELATING OBSERVABLES
TO DYNAMIC CONSIDER PARAMETER STATE

AL(4,15)	OBSERVATION MATRIX RELATING OBSERVABLES TO MEASUREMENT CONSIDER PARAMETER STATE
HPHR(4,4)	NON-FUNCTIONAL IN PRESENT ERROR ANALYSIS PROGRAM
PP(6,6)	POSITION/VELOCITY COVARIANCE MATRIX PRIOR TO PROCESSING A MEASUREMENT
CXXSP(6,12)	CORRELATION BETWEEN POSITION/VELOCITY STATE AND SOLVE-FOR PARAMETERS PRIOR TO PROCESSING A MEASUREMENT
CXUP(6,8)	CORRELATION BETWEEN POSITION/VELOCITY STATE AND DYNAMIC CONSIDER PARAMETERS PRIOR TO PROCESSING A MEASUREMENT
CXVP(6,15)	CORRELATION BETWEEN POSITION/VELOCITY STATE AND MEASUREMENT CONSIDER PARAMETERS PRIOR TO PROCESSING A MEASUREMENT
PSP(12,12)	SOLVE-FOR PARAMETER COVARIANCE MATRIX PRIOR TO PROCESSING A MEASUREMENT
CXSUP(12,8)	CORRELATION BETWEEN SOLVE-FOR PARAMETERS AND DYNAMIC CONSIDER PARAMETERS PRIOR TO PROCESSING A MEASUREMENT
CXSVP(12,15)	CORRELATION BETWEEN SOLVE-FOR PARAMETERS AND MEASUREMENT CONSIDER PARAMETERS PRIOR TO PROCESSING A MEASUREMENT

/STMG / MODE: ERRAN

IGAIN =1, USE GAIN1 SUBROUTINE
 =2, USE GAIN2 SUBROUTINE

/STVEC / MODE: ERRAN, SIMUL

XI(6) INITIAL VEHICLE STATE VECTOR OF ORIGINAL NOMINAL
XF(6) FINAL VEHICLE STATE VECTOR OF ORIGINAL NOMINAL
XB(6) BEGINNING ORIGINAL NOMINAL VEHICLE STATE VECTOR
NDIM1 DIMENSION OF SOLVE-FOR PARAMETER STATE
NDIM2 DIMENSION OF DYNAMIC CONSIDER STATE
NDIM3 DIMENSION OF MEASUREMENT CONSIDER PARAMETER STATE
IAUGIN(24) INPUT AUGMENTATION VECTOR OF ONE'S AND ZERO'S
IAUG(24) AUGMENTATION VECTOR
IAUGDC(8) DYNAMIC CONSIDER AUGMENTATION VECTOR
IAUGMC(15) MEASUREMENT CONSIDER AUGMENTATION VECTOR

/TAREAL/ MODE: NOMNAL, ERRAN, SIMUL

AC(5,10)	ACCURACY LEVELS (UP TO 5) USED IN EACH GUIDANCE EVENT
PHI(3,3)	TARGETING MATRIX
TIMG(10)	TIMES OF EACH GUIDANCE EVENT REFERENCED TO EPOCH-INITIAL TIME, SOI TIME, OR CA TIME
TAR(6,10)	DESIRED VALUES OF TARGET PARAMETERS (UP TO 6 AVAILABLE) FOR EACH GUIDANCE EVENT
DAUX(3)	DESIRED AUXILIARY PARAMETER VALUES OF ITERATE
AAUX(3)	ACTUAL AUXILIARY VALUES OF ITERATE
DTAR(3)	DESIRED TARGET VALUES OF ITERATE
ATAR(3)	ACTUAL TARGET VALUES OF ITERATE
TOL(6,10)	ALLOWABLE TOLERANCES OF TARGET PARAMETERS FOR EACH GUIDANCE EVENT
TOLR(6)	NOT USED IN CURRENT TARGET VERSION
CTOL(6)	TOLERANCES FOR CURRENT EVENT
FAC(3)	SCALING FACTORS USED IN BAD STEP CHECK
TMPR	DAYS BETWEEN PRINTOUTS OF NOMINAL TRAJECTORY
PERV(10)	PERTURBATION SIZE FOR VELOCITY COMPONENTS IN CONSTRUCTING SENSITIVITY MATRICES IN TARGETING EVENTS
DINTG(10)	NOT USED IN CURRENT TARGET VERSION
DT(10)	JULIAN DATES OF TARGET TIMES
DELV(3,10)	EXTERNALLY SUPPLIED VELOCITY CORRECTION OR VELOCITY INCREMENT COMPUTED BY INSERTION DECISION

TRTM	TRAJECTORY TIME (DAYS) REF. TO INJECTION
RIN(6)	CURRENT STATE VECTOR AT I-TH EVENT
TIN	JULIAN DATE AT INJECTION
D1	JULIAN DATE ASSOCIATED WITH RIN ARRAY
DG(10)	JULIAN DATES OF EVENT TIMES
DELTAT	NUMBER OF DAYS INTEGRATION IS TO CONTINUE IF NO OTHER STOPPING CONDITION OCCURS
TMU	GRAVITATIONAL CONSTANT OF TARGET PLANET
RRF(3)	SPACECRAFT POSITION AT END OF INTEGRATION
DELTAV(3)	CORRECTIONS TO BE ADDED TO VELOCITY COMPONENTS FOR NEXT ITERATION
DVMAX(10)	MAXIMUM ALLOWABLE CHANGE IN ANY VELOCITY COMPONENT FOR EACH EVENT
ACKT	TRAJECTORY INTEGRATION ACCURACY
EQECP(3,3)	TRANSFORMATION FROM ECLIPTIC TO EQUATORIAL SYSTEM FOR TARGET PLANE
TIMS	INTERNAL CLOCK TIME AT START OF COMPUTER RUN
SPHFAC(10)	REDUCTION FACTORS FOR TARGET PLANET SPHERE OF INFLUENCE FOR EACH EVENT
RPS(10)	RADIUS OF PROBE IMPACT SPHERE IN KM FOR VARIOUS TYPES OF PROBE EVENTS

/TARINT/ MODE: NOMNAL, ERRAN, SIMUL

NOGYD TOTAL NUMBER OF GUIDANCE EVENTS

KTIM(10) EPOCH TO WHICH GUIDANCE EVENT TIMES ARE REFER-
 ENCED
 =0 EVENT NOT PROCESSED
 =1 INITIAL TIME
 =2 SOI TIME
 =3 CA TIME
 =4 CALENDAR DATE

KTYP(10) TYPE OF GUIDANCE EVENT FOR EACH EVENT
 =-1 TERMINATION EVENT
 =1 TARGETING EVENT
 =2 RETARGETING EVENT
 =3 ORBIT INSERTION EVENT

KMXQ(10) COMPUTE/EXECUTE MODES FOR EACH GUIDANCE EVENT
 =1 COMPUTE VELOCITY CORRECTION ONLY
 =2 EXECUTE VELOCITY CORRECTION ONLY
 =3 COMPUTE AND IMMEDIATELY EXECUTE CORRECTION
 =4 COMPUTE BUT EXECUTE CORRECTION LATER

MDL(10) EXECUTION MODELS FOR EACH GUIDANCE EVENT
 =1 IMPULSIVE
 =2 PULSING ARC

NPAR(10) NUMBER OF TARGET PARAMETERS IN EACH TARGETING
 EVENT

KTAR(6,10) CODES OF TARGET PARAMETERS (UP TO 6) FOR EACH
 TARGETING EVENT OR ORBIT INSERTION OPTION FOR
 EACH INSERTION EVENT

KEYTAR(3) KEY DEFINING DESIRED TARGET PARAMETERS FOR
 CURRENT EVENT

MAT(10) TARGETING MATRIX COMPUTATION CODE FOR EACH TAR-
 GETING EVENT
 =1 COMPUTE TARGETING MATRIX ONLY AT FIRST LEVEL
 =2 COMPUTE TARGETING MATRIX AT EACH STEP

IBADS(10)	BAD STEP FLAGS FOR EACH TARGETING EVENT =1 NEVER USE BAD STEP CHECK =2 USE BAD STEP CHECK AT FINAL LEVEL ONLY =3 USE BAD STEP CHECK AT ALL LEVELS
NOIT(10)	THE NUMBER OF TOTAL ITERATIONS ALLOWED AT THE FIRST AND LAST LEVELS OF TARGETING EVENTS FOR EACH GUIDANCE EVENT
MAXB(10)	THE NUMBER OF BAD STEPS ALLOWED DURING ANY TAR- GETING EVENT
LEVELS	NUMBER OF ACCURACY LEVELS FOR CURRENT EVENT
LEV	CURRENT LEVEL IN CURRENT TARGETING EVENT
NITS	ALLOWABLE NUMBER OF ITERATIONS FOR CURRENT EVENT
MAXBAD	MAXIMUM NUMBER OF BAD ITERATIONS FOR CURRENT EVENT
IBAST	BAD STEP CHECK INDICATOR FOR CURRENT EVENT
MATX	MATRIX COMPUTATION CODE FOR CURRENT TARGETING EVENT (SEE DEFN OF MAT)
ISTART	STAGE OF INITIAL TARGETING =0 NO TARGETING STARTED =1 FIRST PHASE STARTED AND HAVE TARGETING MATRIX =2 SECOND PHASE STARTED AND HAVE MATRIX
IPHASE	PHASE COUNTER FOR CURRENT TARGETING EVENT
NOPHAS	NUMBER OF TARGETING PHASES FOR CURRENT EVENT
ITARM	FLAG TO CONTROL CONSTRUCTION OF TARGETING MATRIX =0 DO NOT COMPUTE TARGETING MATRIX =1 COMPUTE TARGETING MATRIX ON CURRENT ITERATION
IBAD	BAD STEP FLAG FOR CURRENT ACCURACY LEVEL =1 DO NOT CHECK FOR BAD STEP =2 CHECK FOR BAD STEP

ISTOP	STOPPING CONDITION INDICATOR IN SUBROUTINE TARGET =1 STOP ON TIME =2 STOP AT SPHERE OF INFLUENCE =3 STOP AT CLOSEST APPROACH
NOPAR	NUMBER OF TARGET PARAMETERS FOR CURRENT EVENT
KWIT	TERMINATION FLAG =0 CONTINUE RUN =1 TERMINATE RUN
IPRE	CASE FLAG =0 FIRST CASE =1 STACKED CASE
NCPR	NUMBER OF INTEGRATION INCREMENTS BETWEEN PRINT- OUTS OF NOMINAL TRAJECTORY
IFINT(10)	NOT USED IN THIS TRAJECTORY VERSION
KGYD(10)	INDICES OF EVENTS TO BE PROCESSED
KSICA	FLAG INDICATING STAGE OF NOMINAL TRAJECTORY =1 SOI NOT YET INTERSECTED =2 SOI INTERSECTED BUT NO CLOSEST APPROACH =3 CLOSEST APPROACH ALREADY ENCOUNTERED
KUR	INDEX OF CURRENT EVENT
KAXTAR(3)	KEY DEFINING AUXILIARY PARAMETERS FOR CURRENT EVENT
LVLS(10)	NUMBER OF ACCURACY LEVELS TO BE USED ON EACH TAR- GETING EVENT
NOSOI	OUTER TARGETING FLAG =0 NORMAL TARGETING =1 OUTER TARGETING
IPCS(10)	FLAG SPECIFYING PLANETOCENTRIC COORDINATE SYSTEM FOR VARIOUS TYPES OF PROBE EVENTS =0 EQUATORIAL =1 SUBSOLAR ORBIT-PLANE

/TAROM/ MODE: NOMNAL

DBR	DESIRED VALUE OF AUXILIARY TARGET B.R IN KM TO ACHIEVE ACTUAL TARGET PAIRS OF EITHER INCLINATION AND RADIUS OF CLOSEST APPROACH OR RIGHT ASCENSION AND DECLINATION OF PROBE TARGET SITE
DBT	DESIRED VALUE OF AUXILIARY TARGET B.R IN KM TO ACHIEVE ACTUAL TARGET PAIRS OF EITHER INCLINATION AND RADIUS OF CLOSEST APPROACH OR RIGHT ASCENSION AND DECLINATION OF PROBE TARGET SITE
DDCP	ACTUAL TARGET VALUE OF PROBE SITE DECLINATION IN DEG RELATIVE TO PROBE-SPHERE FRAME
DINC	ACTUAL TARGET VALUE OF INCLINATION AT CLOSEST APPROACH IN DEG RELATIVE TO EQUATORIAL FRAME
DRAP	ACTUAL TARGET VALUE OF PROBE SITE RIGHT ASCENSION IN DEG RELATIVE TO PROBE-SPHERE FRAME
DRCA	ACTUAL TARGET VALUE OF RADIUS AT CLOSEST APPROACH IN KM
IAUX	FLAG INDICATING TYPE OF AUXILIARY TARGETING =0 NO AUXILIARY TARGETING =1 AUXILIARY TARGETING WITH INCLINATION AND RADIUS AT CLOSEST APPROACH AS ACTUAL TARGETS =2 AUXILIARY TARGETING WITH RIGHT ASCENSION AND DECLINATION OF PROBE IMPACT SITE AS ACTUAL TARGETS
IINCRA	INDEX OF DESIRED INCLINATION OR PROBE SITE RIGHT ASCENSION IN DTAR ARRAY
IRCADC	INDEX OF DESIRED RADIUS OF CLOSEST APPROACH OR PROBE SITE DECLINATION IN DTAR ARRAY
ITARR	FLAG INDICATING OPERATING MODE OF TAROPT (SAME AS ITARO)
XATAR(3)	ARRAY OF ACTUAL TARGET VALUES (SAME AS ATAR)

/TARVAR/ MODE ERRAN, SIMUL

XTAR(6,10)	DESIRED TARGET VALUES
XTOL(6,10)	TOLERANCES ON TARGET PARAMETERS
XAC(5,10)	ACCURACY LEVELS EMPLOYED IN TARGETING
XPERV(10)	VELOCITY PERTURBATION USED TO COMPUTE TARGETING MATRIX
XDVMAX(10)	MAXIMUM ALLOWABLE VELOCITY CORRECTION
XFAC(10)	SPHERE OF INFLUENCE FACTORS
XDELV(3,10)	NONLINEAR VELOCITY CORRECTION
TGT3(10)	DESIRED TARGET TIMES REFERENCED TO INITIAL TRAJECTORY TIME
LKTAR(6,10)	ARRAY DEFINING TARGET PARAMETERS
LKTP(10)	ARRAY OF TARGET PLANETS
LKLP(10)	ARRAY OF LAUNCH PLANETS
LNPAR(10)	NUMBER OF TARGET PARAMETERS DESIRED
LLVLS(10)	NUMBER OF INTEGRATION ACCURACY LEVELS USED

/TIM / MODE: ERRAN, SIMUL

DATEJ	JULIAN DATE OF INITIAL TRAJECTORY TIME (REFERENCED TO 1950)
TRTM1	INITIAL TRAJECTORY TIME
DELT	TIME INCREMENT
FNTM	FINAL TRAJECTORY TIME
UNIVT	UNIVERSAL TIME
TRTMB	TRAJECTORY TIME AT BEGINNING OF TRAJECTORY

/TMW2 / MODE: SIMUL

T1	EIGENVECTOR EVENT TIMES
T2	PREDICTION EVENT STARTING TIMES
T4	CONIC COMPUTATION EVENT TIMES
T5	QUASI-LINEAR EVENT TIMES
T6	NOT USED
T7	NOT USED

/TRAJCD/ MODE: ERRAN, SIMUL

DTMAX	MAXIMUM TIME INCREMENT FOR WHICH ISTMC IS VALID
ACCND	ACCURACY USED IN NUMERICAL DIFFERENCING IF NDACC INDICATES
DT\$UN	STATE TRANSITION INTEGRATION INTERVAL WHEN THE SUN IS CENTRAL BODY AND ISTM1=1
DTPLAN	STATE TRANSITION INTEGRATION INTERVAL WHEN TARGET PLANET IS CENTRAL BODY AND ISTM1=1
NTMC	NOMINAL TRAJECTORY CODE
ISTMC	STATE TRANSITION MATRIX CODE
ISTM1	ALTERNATE STATE TRANSITION MATRIX CODE
NDACC	NUMERICAL DIFFERENCING ACCURACY CODE

/TRIAR / MODE: ERRAN, SIMUL

DCTP(3)	DECLINATIONS OF 3 MINI-PROBE TAGETS
RATP(3)	RIGHT ASCENSIONS OF 3 MINI-PROBE TARGETS
VTANGM	MINI-PROBE TANGENTIAL VELOCITY
DGSAF	FIXED SPIN AXIS DECLINATION AT RELEASE
RASAF	FIXED SPIN AXIS RIGHT ASCENSION AT RELEASE
SU	STEP SIZE UPPER BOUND IN THE CONTROL SPACE
ACTPP	VMP ACCURACY LEVEL FOR MINI-PROBE TARGETING
FACIR	FRACTION OF SOI INTEGRATED TO BEFORE CONIC PROPAGATION STARTS
XSAVE(6)	STATE VECTOR AT RELEASE
WFLS(3)	WEIGHTING FACTORS FOR TARGET SITES
UCNTRL(5)	TARGET CONTROLS
IPCSK	=1 SUBSOLAR COORDINATE SYSTEM =2 EQUATORIAL COORDINATE SYSTEM
ISAO	SPIN AXIS ORIENTATION FLAG
IPROPI	TRAJECTORY PROPAGATION CODE

/TRJ / MODE: ERRAN, SIMUL

RCA1(6)	STATE AT CLOSEST APPROACH ON ORIGINAL NOMINAL
RCA2(6)	STATE AT CLOSEST APPROACH ON MOST RECENT NOMINAL
RCA3(6)	STATE AT CLOSEST APPROACH ON ACTUAL TRAJECTORY
RSOI1(3)	POSITION AT SPHERE OF INFLUENCE ON ORIGINAL NOMINAL
RSOI2(3)	POSITION AT SPHERE OF INFLUENCE ON MOST RECENT NOMINAL
RSOI3(3)	POSITION AT SPHERE OF INFLUENCE ON ACTUAL TRAJECTORY
VSOI1(3)	VELOCITY AT SPHERE OF INFLUENCE ON ORIGINAL NOMINAL
VSOI2(3)	VELOCITY AT SPHERE OF INFLUENCE ON MOST RECENT NOMINAL
VSOI3(3)	VELOCITY AT SPHERE OF INFLUENCE ON ACTUAL TRAJECTORY
TCA1	TIME AT CLOSEST APPROACH OF ORIGINAL NOMINAL
TCA2	TIME AT CLOSEST APPROACH OF MOST RECENT NOMINAL
TCA3	TIME AT CLOSEST APPROACH OF ACTUAL TRAJECTORY
TSOI1	TIME AT SPHERE OF INFLUENCE OF ORIGINAL NOMINAL
TSOI2	TIME AT SPHERE OF INFLUENCE OF MOST RECENT NOMINAL
TSOI3	TIME AT SPHERE OF INFLUENCE OF ACTUAL TRAJECTORY

BSI1	B ON ORIGINAL NOMINAL
BSI2	B ON MOST RECENT NOMINAL
BSI3	B ON ACTUAL TRAJECTORY
BDSI1	B DOT T ON ORIGINAL NOMINAL
BDSI2	B DOT T ON MOST RECENT NOMINAL
BDSI3	B DOT T ON ACTUAL TRAJECTORY
BORSI1	B DOT R ON ORIGINAL NOMINAL
BORSI2	B DOT R ON MOST RECENT NOMINAL
BORSI3	B DOT R ON ACTUAL TRAJECTORY
ISOI1	SPHERE OF INFLUENCE CODE FOR ORIGINAL NOMINAL
ISOI2	SPHERE OF INFLUENCE CODE FOR MOST RECENT NOMINAL
ISOI3	SPHERE OF INFLUENCE CODE FOR ACTUAL TRAJECTORY
ICA1	CLOSEST APPROACH CODE FOR ORIGINAL NOMINAL
ICA2	CLOSEST APPROACH CODE FOR MOST RECENT NOMINAL
ICA3	CLOSEST APPROACH CODE FOR ACTUAL TRAJECTORY

 /TMTRIX/ MODE: NOMNAL, ERRAN, SIMUL

CHI(3,3) SENSITIVITY MATRIX (TRANSFERRED FOR OUTPUT)

/TPTIN / MODE: ERRAN ,SIMUL

GMUP	GRAVITATIONAL CONSTANT OF TARGET PLANET
RTPS	RADIUS OF PROBE SPHERE
T(3,3)	COORDINATE SYSTEM TRANSFORMATION MATRIX
DJERN	JULIAN DATE OF TARGETING EVENT
DTPRSC RSCRPA(3)	TIME FROM PERIAPSIS TO NOMINAL RELEASE TIME RELEASE POSITION OF S/C
VSCRPA(3)	RELEASE VELOCITY OF S/C
RSCRPM	MAGNITUDE OF RELEASE POSITION VECTOR
CSDCSA	COS OF DECLINATION OF SPIN AXIS
SNDCSA	SIN OF DECLINATION OF SPIN AXIS
CSRASA	COS OF RIGHT ASCENSION OF SPIN AXIS
SNRASA	SIN OF RIGHT ASCENSION OF SPIN AXIS
NNTP	NUMBER OF THE TARGET PLANET
IMIN	INDEX OF THE MINI-PROBE ¹ NEAREST THE S/C AT IMPACT
KKWIT	=1 NO CONVERGENCE IN MINI-PROBE TARGETING =0 CONVERGENCE IN MINI-PROBE TARGETING

/TPTIN / MODE: NOMNAL

AATTP(3)	ARRAY OF ANGLES OF ATTACK IN DEG FOR MINIPROBES AT IMPACT
ACTPP	ACCURACY LEVEL OF VMP MINIPROBE PROPAGATION
CSDCSA	COSINE OF ECLIPTIC DECLINATION OF BUS SPIN AXIS AT RELEASE
CSRASA	COSINE OF ECLIPTIC RIGHT ASCENSION OF BUS SPIN AXIS AT RELEASE
DCTP(3)	ARRAY OF TARGET SITE DECLINATIONS OF MINIPROBES IN DEG RELATIVE TO PLANETOCENTRIC PROBE-SPHERE FRAME
DELTm	MAXIMUM VIRTUAL-MASS PROPAGATION INTERVAL FOR BUS AND MINIPROBES
DJEITP(3)	ARRAY OF JULIAN DATES OF IMPACT FOR MINIPROBES EPOCH 1900
DJERN	JULIAN DATE EPOCH 1900 OF MINIPROBE RELEASE
DTPRSC	TIME INTERVAL IN SEC ON BUS NEAR-PLANET OSCULATING CONIC FROM PERIAPSIS TO RELEASE STATE
FPATP(3)	ARRAY OF FLIGHT PATH ANGLES IN DEG FOR MINIPROBES AT IMPACT
GMUP	GRAVITATIONAL CONSTANT OF TARGET PLANET IN KM^3/SEC^2
IFIN2	FLAG INDICATING OPERATING MODE OF TPPROP =1 MISS-MINIMIZATION IS IN PROCESS--OBTAIN PHI AS A FUNCTION OF UCNTL =2 MISS-MINIMIZATION IS COMPLETE--OBTAIN MINIPROBE IMPACT DATA FOR MINIMUM-MISS RELEASE CONTROLS
IMIN	INDEX OF MINIPROBE WHOSE IMPACT PLANE ASYMPTOTE PIERCE POINT IS NEAREST THAT OF BUS
IPROP	FLAG INDICATING MINIPROBE PROPAGATION MODE TO BE USED IN TPPROP (MAY DIFFER FROM THAT REQUESTED IN IPROPI) =1 CONIC =2 VIRTUAL-MASS
ISAO	FLAG INDICATING SPIN AXIS ORIENTATION MODE

- =1 BOTH SPIN AXIS DECLINATION AND RIGHT
ASCENSION ARE FREE CONTROLS
- =2 SPIN AXIS IS COINCIDENT WITH BUS VELOCITY
VECTOR AT RELEASE
- =3 SPIN AXIS IS NORMAL TO BUS/SUN LINE, PARALLEL
TO ECLIPTIC PLANE, AND WITHIN 90 DEG OF BUS
VELOCITY VECTOR
- =4 SPIN AXIS DECLINATION AND RIGHT ASCENSION
ARE BOTH FIXED

RATP(3)	ARRAY OF TARGET SITE RIGHT ASCENSIONS OF MINIPROBES IN DEG RELATIVE TO PLANETOCENTRIC PROBE-SPHERE FRAME
RSCRHA(3)	HELIOCENTRIC ECLIPTIC POSITION VECTOR OF BUS IN KM AT MINIPROBE RELEASE
RSCRPA(3)	PLANETOCENTRIC ECLIPTIC POSITION VECTOR IN KM ON BUS NEAR-PLANET OSCULATING CONIC AT EQUIVALENT RELEASE STATE
RSCRPM	MAGNITUDE OF PLANETOCENTRIC POSITION VECTOR OF BUS IN KM AT EQUIVALENT CONIC RELEASE STATE
RTPS	RADIUS OF MINIPROBE IMPACT SPHERE IN KM (SAME AS RADIUS OF PROBE IMPACT SPHERE)
SNDCSA	SINE OF ECLIPTIC DECLINATION OF BUS SPIN AXIS AT RELEASE
SNRASA	SINE OF ECLIPTIC RIGHT ASCENSION OF BUS SPIN AXIS AT RELEASE
TRANSF	TRANSFORMATION MATRIX FROM PLANETOCENTRIC ECLIPTIC TO PROBE-SPHERE COORDINATE FRAME
VIMTP(3)	ARRAY OF VELOCITY MAGNITUDES OF MINIPROBES AT IMPACT IN KM/SEC
VSCRHA(3)	HELIOCENTRIC ECLIPTIC VELOCITY VECTOR OF BUS IN KM/SEC AT MINIPROBE RELEASE
VSCRPA(3)	PLANETOCENTRIC ECLIPTIC VELOCITY VECTOR IN KM/SEC ON BUS NEAR-PLANET OSCULATING CONIC AT EQUIVALENT RELEASE STATE
WFLS(3)	ARRAY OF WEIGHTING FACTORS APPLIED TO IMPACT PLANE MISS DISTANCES AT RESPECTIVE MINIPROBE TARGET SITES IN LEAST-SQUARES MISS-MINIMIZATION PROCESS

/VM / MODE: NOMNAL, ERRAN, SIMUL

ALNGTH	LENGTH UNITS PER A.U.
TM	TIME UNITS PER DAY
DELT.P	PRINT INCREMENTS (IN DAYS)
RG(6)	STATE AT CLOSEST APPROACH
DC	JULIAN DATE,EPOCH 1900,AT CLOSEST APPROACH
RSI(3)	POSITION AT SPHERE OF INFLUENCE
VSI(3)	VELOCITY AT SPHERE OF INFLUENCE
DSI	JULIAN DATE, EPOCH 1900, AT SPHERE OF INFLUENCE
RVS(6)	POSITION OF VEHICLE RELATIVE TO VIRTUAL MASS
VMU	GRAVITATIONAL CONSTANT OF VIRTUAL MASS
B	B AT SPHERE OF INFLUENCE
BOT	B DOT T
BOR	B DOT R
DELTH	INCREMENT IN TRUE ANOMALY USED
TIMINT	TOTAL TIME USED
RE(6)	POSITION AND VELOCITY OF EARTH
RTP(6)	POSITION AND VELOCITY OF TARGET PLANET
CAINC	INCLINATION AT CLOSEST APPROACH
RCA	MAGNITUDE OF CLOSEST APPROACH POSITION VECTOR
TACA	TRAJECTORY SEMIMAJOR AXIS WITH RESPECT TO TARGET BODY AT CLOSEST APPROACH TO TARGET BODY

SSS(3)	DIRECTION COSINE VECTOR OF SPACECRAFT SPIN AXIS
NLP	CODE OF LAUNCH PLANET
NBOD	NUMBER OF BODIES USED IN VIRTUAL MASS PROGRAM
NB(11)	CODES OF PLANETS
NTP	CODE OF TARGET PLANET
INPR	PRINT INCREMENTS (IN INCREMENTS)
IPROB	PROBLEM NUMBER
ISPH	SPHERE OF INFLUENCE CODE =0 SPHERE OF INFLUENCE NOT INTERSECTED =1 SPHERE OF INFLUENCE ALREADY ENCOUNTERED
INCMT	TOTAL INCREMENTS USED
IEPHEM	EPHEMERIS CODE
ICL	CLOSEST APPROACH CODE =0 CLOSEST APPROACH NOT ENCOUNTERED =1 CLOSEST APPROACH ALREADY ENCOUNTERED
IPRINT	PRINT CODE =0 OUTPUT INITIAL AND FINAL DATA =1 DO NOT OUTPUT INITIAL AND FINAL DATA
ICL2	CLOSEST APPROACH TERMINATION CODE =0 DO NOT STOP AT CLOSEST APPROACH =1 STOP AT CLOSEST APPROACH

/XXXL / MODE: ERRAN, SIMUL

XSL(24)	SOLVE-FOR PARAMETER LABELS
XU(8)	DYNAMIC CONSIDER PARAMETER LABELS
XV(15)	MEASUREMENT CONSIDER PARAMETER LABELS
XLAB(6)	VEHICLE POSITION/ VELOCITY VECTOR COMPONENT NAMES
XNM(24)	AUGMENTATION PARAMETER LABELS
KPRINT	CORRELATION MATRIX PRINT CODE

/ZERDAT/ MODE: NOMNAL

ZDAT(6)	ZERO ITERATE VECTOR
RP	PARKING ORBIT RADIUS
FI	INJECTION TRUE ANOMALY
PSI1	ANGLE OF FIRST BURN
PSI2	ANGLE OF SECOND BURN
TIM1	TIME INTERVAL OF FIRST BURN
TIM2	TIME INTERVAL OF SECOND BURN
THELS	LONGITUDE OF LAUNCH SITE
PHILS	LATITUDE OF LAUNCH SITE
TI	NOT USED
TF	NOT USED
THEDOT	ROTATION RATE OF LAUNCH PLANET
RPRAT	PARKING ORBIT INVERSE RATE

SIGMAL	NOMINAL LAUNCH AZIMUTH
IZERO	ZERO ITERATION FLAG =0 INITIAL STATE READ IN =1 PLANET-TO-PLANET =2 PLANET-TO-POINT =3 POINT-TO-PLANET =4 POINT-TO-POINT =10 LUNAR TARGETING
KOAST	PARKING ORBIT INDICATOR =-1 SHORT COAST =1 LONG COAST
LTARG	TYPE OF MISSION FOR TARGETING =0 INTERPLANETARY MISSION =1 LUNAR MISSION

 /ZOUT / MODE: NOMNAL

VHPM	MAGNITUDE OF HYPERBOLIC EXCESS VELOCITY AT TARGET BODY (VHP VECTOR)
DPA	DECLINATION OF VHP
RAP	RIGHT ASCENSION OF VHP

5.2 COMMON VARIABLES IN ALPHABETICAL ORDER

IN THIS SECTION ALL VARIABLES APPEARING IN COMMON ARE LISTED AND DEFINED IN ALPHABETICAL ORDER. THE SECOND FIELD SERVES TO IDENTIFY THE BLOCK IN WHICH THE VARIABLE APPEARS.

A(2,3)	PBLK	FTA IMPACT PLANE TRANSFORMATION MATRIX
AALP(20)	SIMCNT	ACTUAL ERROR IN POINTING ANGLE 1
AAUX(3)	TAREAL	ACTUAL AUXILIARY VALUES OF ITERATE
ABET(20)	SIMCNT	ACTUAL ERROR IN POINTING ANGLE 2
AC(5,10)	TAREAL	ACCURACY LEVELS(UP TO 5) USED IN EACH GUIDANCE EVENT
ACC	MISC	ACCURACY FIGURE USED IN VIRTUAL MASS PROGRAM
ACC1	SIM2	ACCURACY USED IN ACTUAL TRAJECTORY
ACCND	TRAJCD	ACCURACY USED IN NUMERICAL DIFFERENCING IF NDACC INDICATES
ACKT	TAREAL	TRAJECTORY INTEGRATION ACCURACY
ADA(3,6)	BAIN	VARIATION MATRIX
ADEVSB(24)	SIM1	ACTUAL DEVIATION IN SOLVE-FOR PARAMETERS AT TRAJECTORY BEGINNING
ADEVX(6)	SIM1	ACTUAL DEVIATION IN THE STATE VECTOR
ADEVXB(6)	SIM1	ACTUAL DEVIATION IN STATE VECTOR AT BEGINNING OF TRAJECTORY
ADEVXS(24)	SIM1	ACTUAL DEVIATION IN SOLVE-FOR PARAMETERS
AINC7	EVENT	ORBIT INSERTION VARIABLES. NON-FUNCTIONAL IN EXISTING PROGRAM
AK(6,4)	STM	KALMAN GAIN CONSTANT FOR POSITION/VELOCITY STATE

AL(4,15)	STM	OBSERVATION MATRIX RELATING OBSERVABLES TO MEASUREMENT CONSIDER PARAMETER STATE
ALNGTH	VM	LENGTH UNITS PER A.U.
AM(4,24)	STM	OBSERVATION MATRIX RELATING OBSERVABLES TO SOLVE-FOR PARAMETER STATE
ANODE7	EVENT	ORBIT INSERTION VARIABLES. NON-FUNCTIONAL IN IN EXISTING PROGRAM
ANG	IMPTAR	TARGET INCLINATION CONVERTED FROM INPUT FORMAT TO VALUE BETWEEN 0 AND 180 DEGREES AND SATISFYING APPROACH ASYMPTOTE CONSTRAINT
ANOIS(4)	SIM1	ACTUAL WHITE NOISE
APRO(20)	SIMCNT	ACTUAL PROPORTIONALITY ERROR
AR(4,4)	SIM1	ACTUAL MEASUREMENT NOISE
ARES(20)	SIMCNT	ACTUAL RESOLUTION ERROR
ATAR(3)	TAREAL	ACTUAL TARGET VALUES OF ITERATE
ATRANS(6)	BAIM	CLOSEST APPROACH STATE
AVARM(12)	SIMCNT	VARIANCE OF ACTUAL MEASUREMENT NOISE
AY(4)	SIM1	ACTUAL MEASUREMENT
AYMEY(4)	SIM1	ACTUAL MEASUREMENT MINUS ESTIMATED MEASUREMENT
B	VM	B AT SPHERE OF INFLUENCE
BCON(3)	LUNART	MAXIMUM STEP SIZES OF CONTROLS
BDR	VM	B DOT R
BDT	VM	B DOT T
BIA(12)	MISC	MEASUREMENT BIASES

BSI1	TRJ	B ON ORIGINAL NOMINAL
BSI2	TRJ	B ON MOST RECENT NOMINAL
BSI3	TRJ	B ON ACTUAL TRAJECTORY
BDTSI1	TRJ	B DOT T ON ORIGINAL NOMINAL
BDTSI2	TRJ	B DOT T ON MOST RECENT NOMINAL
BDTSI3	TRJ	B DOT T ON ACTUAL TRAJECTORY
BDRSI1	TRJ	B DOT R ON ORIGINAL NOMINAL
BDRSI2	TRJ	B DOT R ON MOST RECENT NOMINAL
BDRSI3	TRJ	B DOT R ON ACTUAL TRAJECTORY
CAI	LUNART	DESIRED CLOSEST APPROACH EQUATORIAL INCLINATION
CAINC	VM	INCLINATION AT CLOSEST APPROACH
CHI(3,3)	TMTRX	SENSITIVITY MATRIX(TRANSFERRED FOR OUTPUT)
CMPNM(30)	NAME	COMPONENT NAME
CN(80)	BLK	CONSTANTS USED TO CALCULATE THE ORBITAL ELEMENTS OF THE FIRST FIVE PLANETS (SEE LARGE ARRAY DEFNS IN SECTION 5.3)
CR	PBLK	CAPTURE RADIUS OF TARGET PLANET
CTOL(6)	TAREAL	TOLERANCES FOR CURRENT EVENT
CXSU(24,8)	STM	CORRELATION BETWEEN SOLVE-FOR PARAMETERS AND DYNAMIC CONSIDER PARAMETERS
CXSUB(24,8)	STM	CORRELATION BETWEEN SOLVE-FOR PARAMETERS AND DYNAMIC CONSIDER PARAMETERS AT INITIAL TIME
CXSUG(24,8)	GUI	CONTROL CORRELATION BETWEEN SOLVE-FOR PARAMETERS AND DYNAMIC CONSIDER PARAMETERS
CXSUP(24,8)	GUI	CORRELATION BETWEEN SOLVE-FOR PARAMETERS AND DYNAMIC CONSIDER PARAMETERS PRIOR TO PROCESSING A MEASUREMENT

CXSV(24,15)	STM	CORRELATION BETWEEN SOLVE-FOR PARAMETERS AND MEASUREMENT CONSIDER PARAMETERS
CXSVB(24,15)	STM	CORRELATION BETWEEN SOLVE-FOR PARAMETERS AND MEASUREMENT CONSIDER PARAMETERS AT INITIAL TIME
CXSVG(24,15)	GUI	CONTROL CORRELATION BETWEEN SOLVE-FOR PARAMETERS AND MEASUREMENT CONSIDER PARA- METERS
CXSVP(24,15)	STM	CORRELATION BETWEEN SOLVE-FOR PARAMETERS AND MEASUREMENT CONSIDER PARAMETERS PRIOR TO PROCESSING A MEASUREMENT
CXU(6,8)	STM	CORRELATION BETWEEN POSITION/VELOCITY STATE AND DYNAMIC CONSIDER PARAMETERS
CXUB(6,8)	STM	CORRELATION BETWEEN POSITION/VELOCITY STATE AND DYNAMIC CONSIDER PARAMETERS AT INITIAL TIME
CXUG(6,8)	GUI	CONTROL CORRELATION BETWEEN POSITION/ VELOCITY STATE AND DYNAMIC CONSIDER PARAMETERS
CXUP(6,8)	STM	CORRELATION BETWEEN POSITION/VELOCITY STATE AND DYNAMIC CONSIDER PARAMETERS PRIOR TO PROCESSING A MEASUREMENT
CXV(6,15)	STM	CORRELATION BETWEEN POSITION/VELOCITY STATE AND MEASUREMENT CONSIDER PARAMETERS
CXVB(6,15)	STM	CORRELATION BETWEEN POSITION/VELOCITY STATE AND MEASUREMENT CONSIDER PARMETERS AT INITIAL TIME
CXVG(6,15)	GUI	CONTROL CORRELATION BETWEEN POSITION/ VELOCITY STATE AND MEASUREMENT CONSIDER PARAMETERS
CXVP(6,15)	STM	CORRELATION BETWEEN POSITION/VELOCITY STATE AND MEASUREMENT CONSIDER PARAMETERS PRIOR TO PROCESSING A MEASUREMENT

CXXS(6,24)	STM	CORRELATION BETWEEN POSITION/VELOCITY STATE AND SOLVE-FOR PARAMETERS
CXXSB(6,24)	STM	CORRELATION BETWEEN POSITION/VELOCITY STATE AND SOLVE-FOR PARAMETERS AT INITIAL TIME
CXXSG(6,24)	GUI	CONTROL CORRELATION BETWEEN POSITION/ VELOCITY STATE AND SOLVE-FOR PARAMETERS
CXXSP(6,24)	STM	CORRELATION BETWEEN POSITION/VELOCITY STATE AND SOLVE-FOR PARAMETERS PRIOR TO PROCESSING A MEASUREMENT
D1	TAREAL	JULIAN DATE ASSOCIATED WITH RIN ARRAY
DAB	SIMCNT	BIAS IN SEMI-MAJOR AXIS OF TARGET PLANET
DATEJ	TIME	JULIAN DATE OF INITIAL TRAJECTORY TIME (REFERENCED TO 1950)
DAUX(3)	TAREAL	DESIRED AUXILIARY PARAMETER VALUES OF ITERATE
DC	VM	JULIAN DATE,EPOCH 1900,AT CLOSEST APPROACH
DEB	SIMCNT	BIAS IN ECCENTRICITY OF TARGET PLANET
DECLIN	LUNART	DECLINATION OF APPROACH ASYMPTOTE WITH RESPECT TO LUNAR EQUATOR
DELAXS	CONST3	TARGET PLANET SEMI-MAJOR AXIS FACTOR USED IN NUMERICAL DIFFERENCING
DELECC	CONST3	TARGET PLANET ECCENTRICITY FACTOR USED IN NUMERICAL DIFFERENCING
DELICL	CONST3	TARGET PLANET INCLINATION FACTOR USED IN NUMERICAL DIFFERENCING
DELMA	CONST3	TARGET PLANET MEAN ANOMALY FACTOR USED IN NUMERICAL DIFFERENCING
DELMUP	CONST3	TARGET PLANET GRAVITATIONAL CONSTANT FACTOR USED IN NUMERICAL DIFFERENCING
DELNUS	CONST3	SUN GRAVITATIONAL CONSTANT FACTOR USED IN NUMERICAL DIFFERENCING

DELNOD	CONST3	TARGET PLANET LONGITUDE OF THE ASCENDING NODE FACTOR USED IN NUMERICAL DIFFERENCING
DELPX(3)	EXE	VELOCITY CORRECTION TO BE MODELED AS AN IMPULSE SERIES
DELTAT	TAREAL	NUMBER OF DAYS INTEGRATION IS TO CONTINUE IF NO OTHER STOPPING CONDITION OCCURS
DELTAV(3)	TAREAL	CORRECTIONS TO BE ADDED TO VELOCITY COMPONENTS FOR NEXT ITERATION
DELTH	VM	INCREMENT IN TRUE ANOMALY USED
DELTH	TIME	TIME INCREMENT
DELTP	VM	PRINT INCREMENTS (IN DAYS)
DELV(3,10)	TAREAL	EXTERNALLY SUPPLIED VELOCITY CORRECTION OR VELOCITY INCREMENT COMPUTED BY INSERTION
DELV(3,10)	PBLK	ARRAY OF EXTERNALLY-SUPPLIED VELOCITY CHANGES
DELW	CONST3	TARGET PLANET ARGUMENT OF PERIAPSIS FACTOR USED IN NUMERICAL DIFFERENCING
DG(10)	TAREAL	JULIAN DATES OF EVENT TIMES
DIB	SIMCNT	BIAS IN INCLINATION OF TARGET PLANET
DINTG(10)	TAREAL	NOT USED IN CURRENT TARGET VERSION
DIPX	EXE	JULIAN DATE TRANSFERRED TO EXCUT OR EXCUTS
DMAB	SIMCNT	BIAS IN MEAN ANOMALY
DMUPB	SIMCNT	BIAS IN GRAVITATIONAL CONSTANT OF TARGET PLANET
DMUSB	SIMCNT	BIAS IN GRAVITATIONAL CONSTANT OF SUN
DNCN(3)	CONST	CONSTANTS FROM WHICH DYNAMIC NOISE IS COMPUTED
DNGB	SIMCNT	BIAS IN LONGITUDE OF ASCENDING NODE
DPA	ZOUT	DECLINATION OF VHP

DSI	VM	JULIAN DATE, EPOCH 1900, AT SPHERE OF INFLUENCE
DT(10)	TAREAL	JULIAN DATES OF TARGET TIMES
DTAR(3)	TAREAL	DESIRED TARGET VALUES OF ITERATE
DTAR(3)	LUNART	TARGET VALUES OF SMA,B.T, AND B.R IN LUNAR TARGETING
DTI	PULS	TIME INTERVAL (DAYS) BETWEEN SUCCESSIVE PULSES
DTIME	OVERL	TIME INTERVAL BETWEEN ORBITAL INSERTION DECISION AND EXECUTION
DTMAX	TRAJCD	MAXIMUM TIME INCREMENT FOR WHICH ISTMC IS VALID
DTPLAN	TRAJCD	STATE TRANSITION INTEGRATION INTERVAL WHEN TARGET PLANET IS CENTRAL BODY AND ISTM1=1
DTSUN	TRAJCD	STATE TRANSITION INTEGRATION INTERVAL WHEN THE SUN IS CENTRAL BODY AND ISTM1=1
DUMMYQ(4)	EXE	ARRAY OF EXECUTION ERROR VARIANCES
DUR	PULS	DURATION OF SINGLE PULSE
DV8(3)	EVENT	ORBIT INSERTION VARIABLE. NON-FUNCTIONAL IN EXISTING PROGRAM
DVF(3)	PULS	VELOCITY INCREMENT ADDED ON FINAL PULSE
DVI(3)	PULS	VELOCITY INCREMENT ADDED ON TYPICAL PULSE
DVMAX(10)	TAREAL	MAXIMUM ALLOWABLE CHANGE IN ANY VELOCITY COMPONENT FOR EACH EVENT
DVN(3)	PBLK	COMMANDED VELOCITY CORRECTION TRANSFERRED TO BIAIM
DVRB(3)	PBLK	VELOCITY CORRECTION REQUIRED TO REMOVE AIMPOINT BIAS

DVUP(3)	PBLK	UPDATE VELOCITY CORRECTION
DWB	SIMCNT	BIAS IN ARGUMENT OF PERIAPSIS
ECC7	EVENT	ORBIT INSERTION VARIABLES. NON-FUNCTIONAL IN IN EXISTING PROGRAM
EDEVSM(24)	SIM1	ESTIMATED DEVIATION IN SOLVE-FOR PARAMETERS (FOR ADAPTIVE FILTERING)
EDEVX(6)	SIM1	ESTIMATED DEVIATION IN THE STATE VECTOR
EDEVXM(6)	SIM1	ESTIMATED DEVIATION IN THE STATE VECTOR (FOR ADAPTIVE FILTERING)
EDEVXS(24)	SIM1	ESTIMATED DEVIATION IN SOLVE-FOR PARAMETERS
EIGHT	DPNUM	THE NUMBER EIGHT (8) TO NINE SIGNIFICANT FIGURES. TARGETING MODE ONLY
ELMNT(80)	BLK	CONTAINS THE ORBITAL ELEMENTS OF THE PLANETS (SEE LARGE ARRAY DEFNS IN SECTION 5.3)
EM(2,6)	GUI	VARIATION MATRIX RELATING POSITION /VELOCITY DEVIATIONS TO B.T AND B.R DEVIATIONS
EM1	DPNUM	THE NUMBER 1.E-1 TO NINE SIGNIFICANT FIGURES
EM2	DPNUM	THE NUMBER 1.E-2 TO NINE SIGNIFICANT FIGURES
EM3	DPNUM	THE NUMBER 1.E-3 TO NINE SIGNIFICANT FIGURES
EM4	DPNUM	THE NUMBER 1.E-4 TO NINE SIGNIFICANT FIGURES
EM5	DPNUM	THE NUMBER 1.E-5 TO NINE SIGNIFICANT FIGURES
EM6	DPNUM	THE NUMBER 1.E-6 TO NINE SIGNIFICANT FIGURES

EM7	DPNUM	THE NUMBER 1.E-7 TO NINE SIGNIFICANT FIGURES
EM8	DPNUM	THE NUMBER 1.E-8 TO NINE SIGNIFICANT FIGURES
EM9	DPNUM	THE NUMBER 1.E-9 TO NINE SIGNIFICANT FIGURES
EM13	DPNUM	THE NUMBER 1.E-13 TO NINE SIGNIFICANT FIGURES
EM50	DPNUM	THE NUMBER 1.E-50 TO NINE SIGNIFICANT FIGURES
EMN(15)	BLK	THE CONSTANTS USED TO CALCULATE THE ORBITAL ELEMENTS OF THE MOON
EMU	LUNART	GRAVITATIONAL CONSTANT OF EARTH(KM3/SEC2)
EPS	CONST	OBLIQUITY OF EARTH
EQECP(3,3)	TAREAL	TRANSFORMATION FROM ECLIPTIC TO EQUATORIAL SYSTEM FOR TARGET PLANE
EQLQ(3,3)	LUNART	TRANSFORMATION MATRIX FROM EARTH-EQUATORIAL TO LUNAR EQUATORIAL COORDINTES
EVNM(11)	NAME	EVENT NAME
EXEC(3,3)	PBLK	EXECUTION ERROR COVARIANCE MATRIX
EY(4)	SIM1	ESTIMATED MEASUREMENT
F(44,4)	COM	CONTAINS THE POSITIONS AND VELOCITIES OF THE PLANETS AT A SPECIFIED TIME PLUS THE POSITIONS AND VELOCITIES OF THE SPACE-CRAFT RELATIVE TO THE PLANETS
FAC(3)	TAREAL	SCALING FACTORS USED IN BAD STEP CHECK
FACP	MISC	POSITION FACTOR USED IN NUMERICAL DIFFERENCING
FACV	MISC	VELOCITY FACTOR USED IN NUMERICAL DIFFERENCING
FI	ZERDAY	INJECTION TRUE ANOMALY

FIVE	DPNUM	THE NUMBER FIVE (5) TO NINE SIGNIFICANT FIGURES. TARGETING MODE ONLY
FNTM	TIME	FINAL TRAJECTORY TIME
FOP	CONST2	OFF-DIAGONAL ANNIHILATION VALUE FOR POSITION EIGENVALUES
FOUR	DPNUM	THE NUMBER FOUR (4) TO NINE SIGNIFICANT FIGURES. TARGETING MODE ONLY
FOV	CONST2	OFF-DIAGONAL ANNIHILATION VALUE FOR VELOCITY EIGENVALUES
FS(2,5)	PULS	F-SERIES COEFFICIENTS OF LAUNCH AND TARGET BODIES
G(4,8)	STM	OBSERVATION MATRIX RELATING OBSERVABLES TO DYNAMIC CONSIDER PARAMETER STATE
GA(3,6)	OVERZ	GUIDANCE MATRIX
GG(3)	PULS	GRAVITATIONAL CONSTANTS OF SUN, LAUNCH, AND TARGET BODIES
GS(2,4)	PULS	G-SERIES COEFFICIENTS OF LAUNCH AND TARGET BODIES
H(4,6)	STM	OBSERVATION MATRIX RELATING OBSERVABLES TO POSITION/VELOCITY STATE
HALF	DPNUM	THE NUMBER ONE-HALF (1/2) TO NINE SIGNIFICANT FIGURES
HP7	EVENT	ORBIT INSERTION VARIABLES. NON-FUNCTIONAL IN EXISTING PROGRAM
HPHR(4,4)	STM	NON-FUNCTIONAL IN PRESENT ERROR ANALYSIS PROGRAM
IAMNF	SIMCNT	ACTUAL MEASUREMENT NOISE FLAG
IAUGIN(24)	STVEC	INPUT AUGMENTATION VECTOR OF ONE'S AND ZERO-S
IAUG(24)	STVEC	AUGMENTATION VECTOR
IAUGDC(8)	STVEC	DYNAMIC CONSIDER AUGMENTATION VECTOR

IAUGMC(15)	STVEC	MEASUREMENT CONSIDER AUGMENTATION VECTOR
IBAD	TARINT	BAD STEP FLAG FOR CURRENT ACCURACY LEVEL =1, DO NOT CHECK FOR BAD STEP =2, CHECK FOR BAD STEP
IBADS(10)	TARINT	BAD STEP FLAGS FOR EACH TARGETING EVENT =1 NEVER USE BAD STEP CHEQUE =2 USE BAD STEP CHECK AT FINAL LEVEL ONLY =3 USE BAD STEP CHECK AT ALL LEVELS
IBAG	BAIM	NOT USED
IBARY	CNTRIC	REFERENCE COORDINATE SYSTEM CODE
IBAST	TARINT	BAD STEP CHECK INDICATOR FOR CURRENT EVENT
IBIAS	PBLK	BIASED AIMPOINT GUIDANCE EVENT FLAG = 0 AIMPOINT NOT BIASED =1 AIMPOINT BIASED
ICA1	TRJ	CLOSEST APPROACH CODE FOR ORIGINAL NOMINAL
ICA2	TRJ	CLOSEST APPROACH CODE FOR MOST RECENT NOMINAL
ICA3	TRJ	CLOSEST APPROACH CODE FOR ACTUAL TRAJECTORY
ICDQ3(20)	EVENT	ARRAY OF CODES WHICH DETERMINE WHICH EXECUTION POLICIES ARE TO BE USED IN GUIDANCE EVENTS
ICDT3(20)	EVENT	CODES WHICH DETERMINE WHICH GUIDANCE POLICIES ARE BEING USED
ICL	VM	CLOSEST APPROACH CODE =0 CLOSEST APPROACH NOT ENCOUNTERED =1 CLOSEST APPROACH ENCOUNTERED
ICL2	VM	CLOSEST APPROACH TERMINATION CODE =0 DO NOT STOP AT CLOSEST APPROACH =1 STOP AT CLOSEST APPROACH
ICODE	OVER1	MEASUREMENT CODE
ICORR	MISC	CODE TO DETERMINE WHICH COORDINATE SYSTEM THE INITIAL STATE VECTOR IS INPUT

ICCOORD	CNTRIC	NON-FUNCTIONAL IN ERROR ANALYSIS MODE
IDENS	PBLK	PROBABILITY DENSITY FUNCTION CODE. NON-FUNCT.ONAL
IDNF	MISC	DYNAMIC NOISE FLAG
IEIG	EVENT	CODE USED TO DECIDE IF BOTH POSITION AND VELOCITY EIGENVECTORS ARE REQUESTED
IEPHEM	VM	EPHEMERIS CODE
IEVNT(50)	EVENT	CODES OF EVENTS
IFINT(10)	TARINT	NOT USED IN THIS TRAJECTORY VERSION
IGP	OVERZ	MIDCOURSE GUIDANCE POLICY CODE
IGUID(5,10)	BAIM	ARRAY OF GUIDANCE EVENT CODES
IHYP1	EVENT	HYPERELLIPSOID CODE USED TO DETERMINE IF K=1, K=3, OR BOTH
II	BAIM	GUIDANCE EVENT COUNTER
IIGP	PBLK	MIDCOURSE GUIDANCE POLICY CODE
IIPOL	EVENT	CODE WHICH DETERMINES IF EITHER TWO -VARIABLE OR THREE-VARIABLE B-PLANE GUIDANCE POLICY HAS OCCURRED
IMNF	MISC	MEASUREMENT NOISE FLAG
INC	COM	DETERMINE WHETHER THE ABOVE OPTION IS TO BE USED
INCMNT	COM	NUMBER OF INCREMENTS USED
INCMT	VM	TOTAL INCREMENTS USED
INCPR	COM	SPECIFIES AFTER HOW MANY TIME INCREMENTS PRINT-OUT IS TO OCCUR
INITIAL	CNTRIC	NON-FUNCTIONAL IN ERROR ANALYSIS MODE
INPR	VM	PRINT INCREMENTS (IN INCREMENTS)
INPX	EXE	IMPULSE SERIES CODE

IOPT7	EVENT	ORBIT INSERTION VARIABLE. NON-FUNCTIONAL IN EXISTING PROGRAMS
IPG	COM	PAGE NUMBER
IPHASE	TARINT	PHASE COUNTER FOR CURRENT TARGETING EVENT
IPOL	EVENT	CODE WHICH DETERMINES IF FIXED-TIME-OF- ARRIVAL GUIDANCE EVENT HAS OCCURED
IPQ	BAIM	NOT USED
IPR	COM	A CODE WHICH DETERMINES IF PRINT-OUT IS TO OCCUR AFTER A SPECIFIED NUMBER OF DAYS
IPRE	TARINT	CONTROLS INITIALIZATION OF PROGRAM CONSTANTS IN SUBROUTINE -PRELIM-.
IPRINT	VM	PRINT CODE =0 OUTPUT INITIAL AND FINAL DATA =1 DO NOT OUTPUT DATA
IPROB	VM	PROBLEM NUMBER
IPRT(4)	COM	SPECIFIES PRINT OPTIONS (NOT APPLICABLE TO STEAP TRAJECTORY)
ISOI1	TRJ	SPHERE OF INFLUENCE CODE FOR ORIGINAL NOMINAL
ISOI2	TRJ	SPHERE OF INFLUENCE CODE FOR MOST RECENT NOMINAL
ISOI3	TRJ	SPHERE OF INFLUENCE CODE FOR ACTUAL TRAJECTORY
ISP2	MISC	SPHERE OF INFLUENCE FLAG
ISPH	VM	SPHERE OF INFLUENCE CODE =0 SPHERE OF INFLUENCE NOT INTERSECTED =1 SPHERE OF INFLUENCE INTERSECTED
ISTART	TARINT	STAGE OF INITIAL TARGETING =0 NO TARGETING STARTED =1 FIRST PHASE STARTED -HAVE TARG MATRIX =2 SECOND PHASE STARTED - HAVE TARG MATRIX

ISTM1	TRAJCD	ALTERNATE STATE TRANSITION MATRIX CODE
ISTMC	TRAJCD	STATE TRANSITION MATRIX CODE
ISTOP	TARINT	STOPPING CONDITION INDICATOR IN SUBROUTINE TARGET =1, STOP ON TIME =2, STOP AT SPHERE-OF-INFLUENCE =3, STOP AT CLOSEST APPROACH
ITAG	LUNART	FLAG SPECIFYING STAGE OF TARGETING =1 IN SMA TARGETING =2 IN SHA, INC, RCA TARGETING
ITARM	TARINT	FLAG TO CONTROL CONSTRUCTION OF TARGETING MATRICES =0, DO NOT CALCULATE STATE TRANSITION =1, CALCULATE STATE TRANSITION MATRIX AFTER EACH ITERATION
ITR	MISC	MODE FLAG
ITRAT	COM	IN INTERNAL CODE USED TO DETERMINE HOW MANY ITERATIONS HAVE BEEN ACCOMPLISHED IN THE VIRTUAL MASS PROCEDURE
IX	OVERX	NONLINEAR GUIDANCE CODE
IZERO	ZERDAT	ZERO ITERATION FLAG =0 INITIAL STATE READ IN =1 PLANET TO PLANET =2 PLANET TO POINT =3 POINT TO PLANET =4 POINT TO POINT =10 LUNAR TARGETING
JX	OVERX	GUIDANCE EVENT COUNTER
KAXTAR(3)	TARINT	KEY DEFINING AUXILIARY PARAMETERS FOR CURRENT EVENT
KEYTAR(3)	TARINT	KEY DEFINING DESIRED TARGET PARAMETERS FOR CURRENT EVENT
KGVD(10)	TARINT	FLAG INDICATING GUIDANCE INFORMATION IN CORRESPONDING COLUMNS OF INPUT ARRAYS =1, INFORMATION =0, NO INFORMATION

KL	CON	PROBLEM NUMBER (NOMNAL ONLY)
KMXQ(10)	TARINT	COMPUTE/EXECUTE MODES FOR EACH EVENT =1 COMPUTE VELOCITY CORRECTION ONLY =2 EXECUTE VELOCITY CORRECTION ONLY =3 COMPUTE AND IMMEDIATELY EXECUTE CORRECTIONS =4 COMPUTE BUT EXECUTE CORRECTION LATER
KOAST	ZERDAT	=-1, SHORT COAST. =+1, LONG-COAST
KOUNT	COM	A CODE WHICH SPECIFIES WHETHER PRINT-OUT IS TO OCCUR AFTER THIS TIME INCREMENT
KPRINT	XXXL	CORRELATION MATRIX PRINT CODE
KSICA	TARINT	STOPPING CONDITION INDICATOR IN SUBROUTINE TRJTRY =1, STOP ON TIME =2, STOP AT SPHERE-OF-INFLUENCE =3, STOP AT CLOSEST APPROACH
KTAR(6,10)	TARINT	CODES OF TARGET PARAMETERS (UP TO 6) FOR EACH TARGETING EVENT OR ORBIT INSERTION OPTION FOR EACH INSERTION EVENT
KTIN(10)	TARINT	EPOCH TO WHICH GUIDANCE EVENT TIMES ARE REFERENCED =0 EVENT NOT PROCESSED =1 INITIAL TIME =2 SOI TIME =3 CA TIME =4 CALENDAR DATE
KTYP(10)	TARINT	TYPE OF GUIDANCE EVENT FOR EACH EVENT =-1 TERMINATION EVENT =1 TARGETING EVENT =2 RETARGETING EVENT =3 ORBIT INSERTION EVENT
KMXQ(10)	TARINT	COMPUTE/EXECUTE MODES FOR EACH GUIDANCE EVENT =1 COMPUTE VELOCITY CORRECTION ONLY =2 EXECUTE VELOCITY CORRECTION ONLY =3 COMPUTE AND IMMEDIATELY EXECUTE CORRECTION =4 COMPUTE BUT EXECUTE CORRECTION LATER

KUR	TARINT	FLAG INDICATING WHICH EVENT IS THE CURRENT EVENT
KWIT	TARINT	TERMINATION INDICATOR =0, CONTINUE RUN =1, TERMINATE RUN
LEV	TARINT	CURRENT LEVEL IN CURRENT TARGETING EVENT
LEVELS	TARINT	NUMBER OF ACCURACY LEVELS FOR CURRENT EVENT
LINCT	COM	LINE COUNT (NOMNAL ONLY)
LINPGE	COM	LINES PER PAGE (NOMNAL ONLY)
LKLP(10)	TARVAR	ARRAY OF LAUNCH PLANETS
LKTAR(6,10)	TARVAR	ARRAY DEFINING TARGET PARAMETERS
LKTP(10)	TARVAR	ARRAY OF TARGET PLANETS
LLVLS(10)	TARVAR	NUMBER OF INTEGRATION ACCURACY LEVELS USED
LNPARG(10)	TARVAR	NUMBER OF TARGET PARAMETERS DESIRED
LTARG	ZERDAT	= 0, INTERPLANETARY TARGETING. =1, LUNAR TARGETING
LVLS	TARINT	NUMBER OF ACCURACY LEVELS TO BE USED ON EACH TARGETING EVENT
MAT(10)	TARINT	TARGETING MATRIX COMPUTATION CODE FOR EACH TARGETING EVENT =1 COMPUTE TARGETING MATRIX ONLY AT FIRST LEVEL =2 COMPUTE TARGETING MATRIX AT EACH STEP
MATX	TARINT	MATRIX COMPUTATION CODE FOR CURRENT TARGETING EVENT (SEE DEFN OF MAT)
MAXB(10)	TARINT	THE NUMBER OF BAD STEPS ALLOWED DURING ANY TARGETING EVENT
MAXBAD	TARINT	MAXIMUM NUMBER OF BAD ITERATIONS FOR CURRENT EVENT
MCNTR	MEAS	NUMBER OF MEASUREMENTS HAVING OCCURRED
MCODE(1000)	MEAS	ARRAY OF MEASUREMENT CODES

MDL(10)	TARINT	EXECUTION MODELS FOR EACH GUIDANCE EVENT =1 IMPULSIVE =2 PULSING ARC
MNCN(12)	CONST	MEASUREMENT NOISE CONSTANTS
MNNAME(12,3)	NAME	MEASUREMENT NAME
MONTH(12)	PRT	NAMES OF MONTHS
NAE	EVENT	ADAPTIVE FILTERING EVENTS HAVING OCCURRED.
NAF6(20)	EVENT	ARRAY OF ADAPTIVE FILTERING EVENT CODES. NON-FUNCTIONAL IN EXISTING PROGRAM
NAFC	OVER1	ADAPTIVE FILTER FLAG
NB(11)	VM	CODES OF PLANETS
NB1(11)	SIM2	ARRAY OF PLANET CODES IN ACTUAL TRAJECTORY
NBOD	VM	NUMBER OF BODIES USED IN VIRTUAL MASS PROGRAM
NBOD1	SIM2	NUMBER OF BODIES IN ACTUAL TRAJECTORY
NBODY	COM	EQUAL TO 4*NBODYI-3
NBODYI	COM	NUMBER OF BODIES CONSIDERED IN VIRTUAL MASS TRAJECTORY
NCPR	TARINT	NUMBER OF INTEGRATION INCREMENTS BETWEEN PRINT-OUTS OF NOMINAL TRAJECTORY
NDACC	TRAJCD	NUMERICAL DIFFERENCING ACCURACY CODE
NDIM1	STVEC	DIMENSION OF SOLVE-FOR PARAMETER STATE
NDIM2	STVEC	DIMENSION OF DYNAMIC CONSIDER STATE
NDIM3	STVEC	DIMENSION OF MEASUREMENT CONSIDER PARAMETER STATE
NEV	EVENT	NUMBER OF EVENTS
NEV1	EVENT	TOTAL NUMBER OF EIGENVECTOR EVENTS
NEV2	EVENT	TOTAL NUMBER OF PREDICTION EVENTS

NEV3	EVENT	TOTAL NUMBER OF GUIDANCE EVENTS
NEV4	EVENT	TOTAL NUMBER OF -COMCON- EVENTS
NEV5	EVENT	TOTAL NUMBER OF QUASI-LINEAR FILTERING EVENTS
NEV6	EVENT	TOTAL NUMBER OF ADAPTIVE FILTERING EVENTS. NON-FUNCTIONAL IN EXISTING PROGRAM
NEV7	EVENT	ORBIT INSERTION VARIABLES. NON-FUNCTIONAL IN IN EXISTING PROGRAM
NEV8	EVENT	ORBIT INSERTION VARIABLE. NON-FUNCTIONAL IN EXISTING PROGRAMS
NEV9	EVENT	ORBIT INSERTION VARIABLES. NON-FUNCTIONAL IN IN EXISTING PROGRAM
NEV10	EVENT	ORBIT INSERTION VARIABLES. NON-FUNCTIONAL IN IN EXISTING PROGRAM
NEV11	EVENT	ORBIT INSERTION VARIABLES. NON-FUNCTIONAL IN IN EXISTING PROGRAM
NGE	EVENT	NUMBER OF GUIDANCE EVENTS HAVING OCCURRED
NINETY	DPNUM	THE NUMBER NINETY (90) TO NINE SIGNIFICANT FIGURES. TARGETING MODE ONLY—
NITS	TARINT	ALLOWABLE NUMBER OF ITERATIONS FOR CURRENT EVENT
NLP	VM	CODE OF LAUNCH PLANET
NMN	MEAS	TOTAL NUMBER OF MEASUREMENTS
NO(11)	BLK	AN ARRAY OF PLANET CODES BEING USED TO GENERATE THE VIRTUAL MASS TRAJECTORY
NOGYD	TARINT	TOTAL NUMBER OF GUIDANCE EVENTS
NOIT(10)	TARINT	THE NUMBER OF TOTAL ITERATIONS ALLOWED AT THE FIRST AND LAST LEVELS OF TARGETING EVENTS FOR EACH GUIDANCE EVENT
NOPAR	TARINT	NUMBER OF TARGET PARAMETERS FOR CURRENT EVENT

NOPHAS	TARINT	NUMBER OF TARGETING PHASES FOR CURRENT EVENT
NOSOI	TARINT	OUTER TARGETING FLAG =0 NORMAL TARGETING =1 OUTER TARGETING
NPE	EVENT	NUMBER OF PREDICTION EVENTS HAVING OCCURRED
NPUL	PULS	NUMBER OF PULSES IN PULSING ARC
NQE	EVENT	QUASI-LINEAR FILTERING EVENTS HAVING OCCURRED
NR	OVER1	NUMBER OF ROWS IN THE OBSERVATION MATRIX
NST	CONST	NUMBER OF STATIONS TO BE USED (MAXIMUM 3)
NTMC	TRAJCD	NOMINAL TRAJECTORY CODE
NTP	VM	CODE OF TARGET PLANET
OMEGA	CONST	EARTH-S ROTATION RATE
ONE	DPNUM	THE NUMBER ONE (1) TO NINE SIGNIFICANT FIGURES
OTAR(3)	LUNART	DESIRED VALUES OF SMA,RCA, AND INC
P(6,6)	STM	POSITION/VELOCITY COVARIANCE
PB(6,6)	STM	POSITION/VELOCITY COVARIANCE AT INITIAL TIME
P7	EVENT	ORBIT INSERTION VARIABLES. NON-FUNCTIONAL IN EXISTING PROGRAM
PCON(3)	LUNART	PERTURBATIONS IN CONTROLS(ALPHA,DELTA, THETA)
PERP7	EVENT	ORBIT INSERTION VARIABLES. NON-FUNCTIONAL IN EXISTING PROGRAM.
PERV(10)	TAREAL	PERTURBATIONS SIZE FOR VELOCITY COMPONENTS IN CONSTRUCTING SENSITIVITY MATRICES IN TARGETING EVENTS
PG(6,6)	GUI	POSITION/VELOCITY CONTROL COVARIANCE

PHI(3,3)	TAREAL	TARGETING MATRIX. REQ-D ONLY IF ISTART=1,2
PHI(6,6)	STM	POSITION/VELOCITY STATE TRANSITION MATRIX
PHI2(3,3)	BAIM	INVERSE OF VARIATION MATRIX PARTITION
PHILS	ZERDAT	LATITUDE OF LAUNCH SITE
PI	COM	THE VALUE OF THE MATHEMATICAL CONSTANT PI
PLANET(11)	PRT	NAMES OF PLANETS
PMASS(11)	BLK	GRAVITATIONAL CONSTANTS OF PLANETS IN A.U.**3/DAY**2
POI	PBLK	PROBABILITY OF IMPACT
PP(6,6)	STM	POSITION/VELOCITY COVARIANCE MATRIX PRIOR TO PROCESSING A MEASUREMENT
PROBI	BAIM	ALLOWABLE PROBABILITY OF IMPACT
PS(24,24)	STM	SOLVE-FOR PARAMETER COVARIANCE
PSB(24,24)	STM	SOLVE-FOR PARAMETER COVARIANCE AT INITIAL TIME
PSG(24,24)	GVI	SOLVE-FOR PARAMETER CONTROL COVARIANCE
PSI1	ZERDAT	ANGLE OF FIRST BURN
PSI2	ZERDAT	ANGLE OF SECOND BURN
PSP(24,24)	STM	SOLVE-FOR PARAMETER COVARIANCE MATRIX PRIOR TO PROCESSING A MEASUREMENT
PSTAR	PBLK	NOMINAL PROBABILITY DENSITY FUNCTION EVALUATED AT TARGET PLANET CENTER
PULMAG	PULS	THRUST MAGNITUDE OF PULSING ENGIN
PULMAS	PULS	NOMINAL MASS OF SPACECRAFT DURING PULSING ARC
PULT	PULS	TOTAL TIME INTERVAL OF PULSING ARC

Q(6,6)	STM	DYNAMIC NOISE COVARIANCE MATRIX
QK(6,6)	EXE	EFFECTIVE EXECUTION ERROR COVARIANCE MATRIX
R(4,4)	STM	MEASUREMENT NOISE COVARIANCE MATRIX
RAP	ZOUT	RIGHT ASCENSION OF VHP
RAD	COM	THE NUMBER OF DEGREES PER RADIAN
RADIUS(11)	BLK	THE RADIUS OF A GIVEN PLANET IN A.U.
RC(6)	VM	STATE AT CLOSEST APPROACH
RCA	VM	MAGNITUDE OF CLOSEST APPROACH POSITION VECTOR
RCA	LUNART	RADIUS OF CLOSEST APPROACH TO MOON (DESIRED)
RCA1(6)	TRJ	STATE AT CLOSEST APPROACH ON ORIGINAL NOMINAL
RCA2(6)	TRJ	STATE AT CLOSEST APPROACH ON MOST RECENT NOMINAL
RCA3(6)	TRJ	STATE AT CLOSEST APPROACH ON ACTUAL TRAJECTORY
RE(6)	VM	POSITION AND VELOCITY OF EARTH
RES(4)	SIM1	RESIDUAL
RF(6)	OVER	FINAL TARGETED NOMINAL STATE VECTOR
RF(6)	OVERZ	FINAL TARGETED STATE VECTOR
RF1(6)	OVER	FINAL MOST RECENT NOMINAL STATE VECTOR
RF1(6)	OVERZ	FINAL MOST RECENT NOMINAL STATE VECTOR
RI(6)	OVER1	INITIAL TARGETED NOMINAL STATE VECTOR
RI(6)	OVERR	STATE VECTOR AT EVENT TIME
RI(6)	LUNART	GEOCENTRIC STATE OF S/C AT LUNAR SOI
RI1	OVER1	INITIAL MOST RECENT NOMINAL STATE VECTOR

RIN(6)	TAREAL	CURRENT STATE VECTOR AT I-TH EVENT
RK(2,3)	PULS	POSITION VECTORS OF LAUNCH AND TARGET PLANETS AT IMPULSIVE TIME(MIDPOINT OF PULSING ARC)
RMASS(11)	BLK	THE RELATIVE GRAVITATIONAL CONSTANT OF A STATED PLANET WITH RESPECT TO THE SUN
RME(6)	LUNART	GEOCENTRIC STATE OF CENTER OF MOON IN ECLIPTIC COORDINATES AT TSI
RMQ(6)	LUNART	GEOCENTRIC STATE OF CENTER OF MOON AT TSI IN EQUATORIAL COORDINATES
RP	ZERDAT	PARKING ORBIT RADIUS
RPE	LUNART	RADIUS OF EARTH PARKING ORBIT
RPRAT	ZERDAT	PARKING ORBIT INVERSE RATE
RRF(3)	TAREAL	SPACECRAFT POSITION AT END OF INTEGRATION, USED IN BROKEN PLANE TARGETING
RSI(3)	VM	POSITION AT SPHERE OF INFLUENCE
RSI(6)	LUNART	SELENOCENTRIC STATE OF S/C AT LUNAR SOI
RSOI1(3)	TRJ	POSITION AT SPHERE OF INFLUENCE ON ORIGINAL NOMINAL
RSOI2(3)	TRJ	POSITION AT SPHERE OF INFLUENCE ON MOST RECENT NOMINAL
RSOI3(3)	TRJ	POSITION AT SPHERE OF INFLUENCE ON ACTUAL TRAJECTORY
RTP(6)	VM	POSITION AND VELOCITY OF TARGET PLANET
RVS(6)	VM	POSITION OF VEHICLE RELATIVE TO VIRTUAL MASS
S(24,4)	STM	KALMAN GAIN CONSTANT FOR SOLVE-FOR PARAMETERS
SAL(3)	CONST	ALTITUDES OF STATIONS
SIGALP	EVENT	VARIANCE OF ERROR IN POINTING ANGLE 1

SIGBET	EVENT	VARIANCE OF ERROR IN POINTING ANGLE 2
SIGMAL	ZERDAT	NOMINAL LAUNCH AZIMUTH
SIGPRO	EVENT	VARIANCE OF PROPORTIONALITY ERROR
SIGRES	EVENT	VARIANCE OF RESOLUTION ERROR
SLAT(3)	CONST	LATITUDES OF STATIONS
SLB(9)	SIMCNT	BIASES IN STATION LOCATION CONSTANTS
SLON(3)	CONST	LONGITUDES OF STATIONS
SMA	LUNART	SIMI-MAJOR AXIS OF LUNAR HYPERBOLA (DESIRED)
SMJR(18)	BLK	CONSTANTS USED TO CALCULATE THE SEMI-MAJOR AXES OF THE PLANETS
SPHERE(11)	BLK	THE SPHERES OF INFLUENCE OF THE PLANETS IN A.U.
SPHFAC(10)	TAREAL	REDUCTION FACTORS FOR TARGET PLANET SPHERE OF INFLUENCE FOR EACH EVENT
SSS(3)	VM	DIRECTION COSINES VECTOR OF SPACECRAFT SPIN AXIS
ST(50)	BLK	CONSTANTS USED TO CALCULATE THE ORBITAL ELEMENTS OF THE LAST FOUR PLANETS (SEE LARGE ARRAY DEFNS IN SECTION 5.3)
T	BLK	TRAJECTORY TIME IN DAYS
T1(20)	TMW2	EIGENVECTOR EVENT TIMES
T2(20)	TMW2	PREDICTION EVENT STARTING TIMES
T3(10)	BAIM	ARRAY OF GUIDANCE EVENT TIMES
T4(20)	TMW2	CONIC COMPUTATION EVENT TIMES
T5(20)	TMW2	QUASI-LINEAR EVENT TIMES

T6(20)	TMW2	NOT USED
T7	TMW2	NOT USED
TACA	VM	TRAJECTORY SEMIMAJOR AXIS WITH RESPECT TO TARGET BODY AT CLOSEST APPROACH TO TARGET BODY
TAR(6,10)	TAREAL	DESIRED VALUES OF TARGET PARAMETERS (UP TO 6 AVAILABLE) FOR EACH GUIDANCE EVENT
TAU7	EVENT	ORBIT INSERTION VARIABLES. NON-FUNCTIONAL IN EXISTING PROGRAM
TCA	LUNART	J.D. OF TIME AT LUNAR CLOSEST APPROACH (DESIRED)
TCA1	TRJ	TIME AT CLOSEST APPROACH OF ORIGINAL NOMINAL
TCA2	TRJ	TIME AT CLOSEST APPROACH OF MOST RECENT NOMINAL
TCA3	TRJ	TIME AT CLOSEST APPROACH OF ACTUAL TRAJECTORY
TEN	DPNUM	THE NUMBER TEN (10) TO NINE SIGNIFICANT FIGURES. TARGETING MODE ONLY
TEV(50)	EVENT	TIMES OF EVENTS
TEVN	OVER1	TIME OF CURRENT EVENT
TEVN	OVERR	EVENT TIME
TF	ZERDAT	NOT USED
TG	GUI	TRAJECTORY TIME AT MOST RECENT GUIDANCE EVENT
TGT3(10)	TARVAR	DESIRED TARGET TIMES REFERENCED TO INITIAL TRAJECTORY TIME
THEDOT	ZERDAT	ROTATION RATE OF LAUNCH PLANET
THELS	ZERDAT	LONGITUDE OF LAUNCH SITE

THREE	DPNUM	THE NUMBER THREE (3) TO NINE SIGNIFICANT FIGURES
TI	ZERDAT	NOT USED
TIMG(10)	TAREAL	TIMES OF EACH GUIDANCE EVENT REFERENCED TO EPOCH-INITIAL TIME, SOI TIME, OR CA TIME
TIM1	ZERDAT	TIME OF FIRST BURN
TIM2	ZERDAT	TIME OF SECOND BURN
TIMINT	VM	TOTAL TIME USED
TIMS	TAREAL	INTERNAL CLOCK TIME AT START OF COMPUTER RUN
TIN	TAREAL	JULIAN DATE AT INJECTION
TINJ	BAIN	INJECTION TIME
TM	VM	TIME UNITS PER DAY
TMN(1000)	MEAS	TIMES OF MEASUREMENTS
TMPR(3)	BAIN	MOST RECENT TARGET STATE
TNOMB(3)	BAIN	NOMINAL B-PLANE TARGET STATE
TNOMC(7)	BAIN	NOMINAL CLOSEST APPROACH TARGET STATE
TMU	TAREAL	GRAVITATIONAL CONSTANT OF TARGET PLANET
TMU	LUNART	GRAVITATIONAL CONSTANT OF MOON(KM3/SEC2)
TOL(6,10)	TAREAL	ALLOWABLE TOLERANCES OF TARGET PARAMETERS FOR EACH GUIDANCE EVENT
TRTM	TAREAL	TRAJECTORY TIME (DAYS) REF. TO INJECTION
TOL(6,10)	TAREAL	TOLERANCES OF TARGET PARAMETERS FOR EACH TARGETING EVENT
TOLR(6)	TAREAL	NOT USED IN CURRENT TARGET VERSION

TPT2(20)	EVENT	ARRAY OF TIMES TO WHICH A PREDICTION IS MADE
TRTM	TAREAL	TRAJECTORY TIME ON NOMINAL TRAJECTORY
TRTM1	TIME	INITIAL TRAJECTORY TIME
TRTMB	TIME	TRAJECTORY TIME AT BEGINNING OF TRAJECTORY
TSI	LUNART	PROJECTED J.O. AT SOI INERTSECTION
TSOI1	TRJ	TIME AT SPHERE OF INFLUENCE OF ORIGINAL NOMINAL
TSOI2	TRJ	TIME AT SPHERE OF INFLUENCE OF MOST RECENT NOMINAL
TSOI3	TRJ	TIME AT SPHERE OF INFLUENCE OF ACTUAL TRAJECTORY
TSPH	LUNART	RADIUS OF LUNAR SOI (KM)
TTIM1	SIMCNT	FIRST TIME USED FOR UNMODELLED ACCELERATION
TTIM2	SIMCNT	SECOND TIME USED FOR UNMODELLED ACCELERATION
TTOL(3)	LUNART	ALLOWABLE TOLERANCES IN SMA,B.T,B.R
TWO	DPNUM	THE NUMBER TWO (2) TO NINE SIGNIFICANT FIGURES
TWOPI	DPNUM	THE MATHEMATICAL CONSTANT 2.*PI
TXU(6,8)	STM	STATE TRANSITION MATRIX PARTITION ASSOCIATED WITH DYNAMIC CONSIDER PARAMETERS
TXXS(6,24)	STM	STATE TRANSITION MATRIX PARTITION ASSOCIATED WITH SOLVE-FOR PARAMETERS
UO(8,8)	STM	DYNAMIC CONSIDER PARAMETER COVARIANCE MATRIX
UNIVT	TIME	UNIVERSAL TIME
UNMAC(3,3)	SIMCNT	UNMODELLED ACCELERATION
UST(3)	CONST2	DIRECTION COSINES OF 3 REFERENCE STARS

V(16,7)	COM	AN ARRAY WHICH STORES PERTINANT VECTORS USED IN THE CALCULATION OF THE VIRTUAL MASS TRAJECTORY (SEE LARGE ARRAY DEFNS IN SECTION 5.3)
V0(15,15)	STM	MEASUREMENT CONSIDER PARAMETER COVARIANCE MATRIX
VHPM	ZOUT	MAGNITUDE OF HYPERBOLIC EXCESS VELOCITY AT TARGET BODY (VHP VECTOR)
VINF	BAIM	HYPERBOLIC EXCESS VELOCITY
VK(2,3)	PULS	VELOCITY VECTORS OF LAUNCH AND TARGET PLANETS AT IMPULSIVE TIME (MIDPOINT OF PULSING ARC)
VMU	VM	GRAVITATIONAL CONSTANT OF VIRTUAL MASS
VSI(3)	VM	VELOCITY AT SPHERE OF INFLUENCE
VSOI1(3)	TRJ	VELOCITY AT SPHERE OF INFLUENCE ON ORIGINAL NOMINAL
VSOI2(3)	TRJ	VELOCITY AT SPHERE OF INFLUENCE ON MOST RECENT NOMINAL
VSOI3(3)	TRJ	VELOCITY AT SPHERE OF INFLUENCE ON ACTUAL TRAJECTORY
VST(3)	CONST2	DIRECTION COSINES OF 3 REFERENCE STARS
W(17)	SIM1	ACTUAL DYNAMIC NOISE
WST(3)	CONST2	DIRECTION COSINES OF 3 REFERENCE STARS
XAC(5,10)	TARVAR	ACCURACY LEVELS EMPLOYED IN TARGETING
XB(6)	STVEC	BEGINNING ORIGINAL NOMINAL VEHICLE STATE VECTOR
XBDT	SAVVAL	ORIGINAL VALUE OF B.T IN N-L GUIDANCE
XBDR	SAVVAL	ORIGINAL VALUE OF B.R IN N-L GUIDANCE
XDC	SAVVAL	ORIGINAL VALUE OF DC IN N-L GUIDANCE
XDELV(3,10)	TARVAR	NONLINEAR VELOCITY CORRECTION

XDSI	SAVVAL	ORIGINAL VALUE OF TSI IN N-L GUIDANCE
XDVMAX(10)	TARVAR	MAXIMUM ALLOWABLE VELOCITY CORRECTION
XF(6)	STVEC	FINAL VEHICLE STATE VECTOR OF ORIGINAL NOMINAL
XF1(6)	SIM1	FINAL STATE VECTOR OF MOST RECENT NOMINAL TRAJECTORY
XFAC(10)	TARVAR	SPHERE OF INFLUENCE FACTORS
XG(6)	GUI	STATE VECTOR AT TIME OF LAST GUIDANCE EVENT
XI(6)	STVEC	INITIAL VEHICLE STATE VECTOR OF ORIGINAL NOMINAL
XI1(6)	SIM1	INITIAL STATE VECTOR OF MOST RECENT NOMINAL
XIN(6)	OVERX	STATE VECTOR TRANSFERRED TO NONLIN
XLAB(6)	XXXL	VEHICLE POSITION/VELOCITY VECTOR COMPONENT NAMES
XLAM(2,2)	PBLK	PROJECTION OF TARGET CONDITION COVARIANCE MATRIX INTO THE IMPACT PLANE
XLAMI(2,2)	PBLK	INVERSE OF XLAM(2,2)
XMUS(2)	PBLK	NOMINAL IMPACT PLANE TARGET STATE
XNM(24)	XXXL	AUGUMENTATION PARAMETER LABELS
XP(6)	BLK	THE POSITION AND VELOCITY OF A PLANET IN HELIOCENTRIC ECLIPTIC COORDINATES
XPERV(10)	TARVAR	VELOCITY PERTURBATION USED TO COMPUTE TARGETING MATRIX
XRC(6)	SAVVAL	ORIGINAL VALUE OF RC IN N-L GUIDANCE
XRSI(3)	SAVVAL	ORIGINAL VALUE OF RSI IN N-L GUIDANCE

XSL(24)	XXXL	SOLVE-FOR PARAMETER LABELS
XTAR(6,10)	TARVAR	DESIRED TARGET VALUES
XTOL(6,10)	TARVAR	TOLERANCES ON TARGET PARAMETERS
XU(8)	XXXL	DYNAMIC CONSIDER PARAMETER LABELS
XV(15)	XXXL	MEASUREMENT CONSIDER PARAMETER LABELS
XVSI(3)	SAVVAL	ORIGINAL VALUE OF VSI IN N-L GUIDANCE
XXIN(6)	EXE	STATE VECTOR TRANSFERRED TO EXCUT OR EXCUTS
Z(17)	SIM1	ACTUAL STATE VECTOR
ZDAT(6)	ZERDAT	ZERO ITERATE VECTOR IF IZERO=0 ZDAT(1-6)= INITIAL STATE =2,3,4, ZDAT(1-3)= INITIAL POSITION ZDAT(4-6)= FINAL POSITION
ZERO	DPNUM	THE NUMBER ZERO (0) TO NINE SIGNIFICANT FIGURES
ZI(17)	SIM1	INITIAL ACTUAL STATE VECTOR
ZF(6)	SIM1	FINAL ACTUAL STATE VECTOR AFTER ADDING THE EFFECT OF UNMODELED ACCELERATION

5.3 Large Array Definitions

In this section large arrays appearing in COMMON will be displayed. The arrays depicted are frequently referenced in trajectory propagation subroutines in STEAP; hence the programmer studying such subroutines will find the following tables extremely useful.

Tables 5.1 to 5.5 describe arrays containing planetary ephemeris constants. The values actually stored in these arrays may be found in the documentation for BLOCK DATA. Tables 5.6 through 5.8 contain variables used in the virtual mass propagation procedure. Discussions of these variables may be found in VMP, EPHEM, ORB, and similar routines.

Constant	1	Ω	$\tilde{\omega}$	e	M	a	ω	E	a_0	a_1
Mercury	1	2	3	4	5	6	7	8	1	2
Venus	9	10	11	12	13	14	15	16	3	4
Earth	17	18	19	20	21	22	23	24	5	6
Mars	25	26	27	28	29	30	31	32	7	8
Jupiter	33	34	35	36	37	38	39	40	9	10
Saturn	41	42	43	44	45	46	47	48	11	12
Uranus	49	50	51	52	53	54	55	56	13	14
Neptune	57	58	59	60	61	62	63	64	15	16
Pluto	65	66	67	68	69	70	71	72	17	18
Moon	73	74	75	76	77	78	79	80		

Table 5.1 ELMNT Array -- Conic Elements

Table 5.2 SMJR Array

Constant	i_0	i_1	i_2	i_3	Ω_0	Ω_1	Ω_2	Ω_3	$\tilde{\omega}_0$	$\tilde{\omega}_1$	$\tilde{\omega}_2$	$\tilde{\omega}_3$	e_0	e_1	e_2	e_3	M_0	M_1	M_2	M_3
Mercury	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Venus	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
Earth	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60
Mars	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80

Table 5.3 CN Array -- Inner Planet Constants

Constant	i_0	i_1	Ω_0	Ω_1	$\tilde{\omega}_0$	$\tilde{\omega}_1$	e_0	e_1	M_0	M_1
Jupiter	1	2	3	4	5	6	7	8	9	10
Saturn	11	12	13	14	15	16	17	18	19	20
Uranus	21	22	23	24	25	26	27	28	29	30
Neptune	31	32	33	34	35	36	37	38	39	40
Pluto	41	42	43	44	45	46	47	48	49	50

Table 5.4 ST Array -- Outer Planet Constants

Constant	0	1	2	3	0	1	2	3	L_0	L_1	L_2	L_3	i	e	a
Moon	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Table 5.5 EMN Array -- Lunar Constants

F-Array				Note:
x_1	y_1	z_1	r_1	
\dot{x}_1	\dot{y}_1	\dot{z}_1	v_1	
\dot{x}_{s1}	\dot{y}_{s1}	\dot{z}_{s1}	\dot{r}_{s1}	
\dot{x}_{s1}	\dot{y}_{s1}	\dot{z}_{s1}	v_{s1}	
x_2	y_2	z_2	r_2	
\dot{x}_2	\dot{y}_2	\dot{z}_2	v_2	
x_{s2}	y_{s2}	z_{s2}	r_{s2}	
\dot{x}_{s2}	\dot{y}_{s2}	\dot{z}_{s2}	v_{s2}	
.	.	.	.	
.	.	.	.	
.	.	.	.	

Subscript i indicates component is i -th body referenced to inertial coordinate system
Subscript si indicates component is spacecraft referenced to i -th body

Table 5.6 F-Array -- Ephemeris Data

TRG(1)	$\cos E$	TRG(5)	$\cos i$	TRG(9)	$\cos \omega$	TRG(13)	$\cos(\omega + f)$
TRG(2)	$\sin E$	TRG(6)	$\sin i$	TRG(10)	$\sin \omega$	TRG(14)	$\sin(\omega + f)$
TRG(3)	$\cos f$	TRG(7)	$\cos Q$	TRG(11)	$\cos \omega$		
TRG(4)	$\sin f$	TRG(8)	$\sin Q$	TRG(12)	$\sin \omega$		

Table 5.7 TRG Array -- Trigonometric Functions

I \ J	1	2	3	4	5	6	7
1	$(t_e)_{dim}, t_B$	$(x_{se})_{dim}, x_{sB}$	$(y_{se})_{dim}, y_{sB}$	$(z_{se})_{dim}, z_{sB}$	$\omega (deg/t), \omega (rad/t)$	D	$\mu, -\mu$
2	t_e	x_{se}	y_{se}	z_{se}	$(t_F)_{dim}, t_F$	$(r_{1s_F})_{dim}, r_{1s_F}$	$(r_{2s_F})_{dim}, r_{2s_F}$
3	$(t_{eph_e})_{dim}, t_{eph_B}$	$(\dot{x}_{se})_{dim}, \dot{x}_{sB}$	$(\dot{y}_{se})_{dim}, \dot{y}_{sB}$	$(\dot{z}_{se})_{dim}, \dot{z}_{sB}$	$(\Delta t_P)_{dim}, \Delta t_P$	C_2	
4	t_{eph_e}	\dot{x}_{se}	\dot{y}_{se}	\dot{z}_{se}	$\frac{\Delta r}{r}$		ωD (velocity)
5	$(\mu_{ve})_{dim}, \mu_{vB}$	$(x_{ve})_{dim}, x_{vB}$	$(y_{ve})_{dim}, y_{vB}$	$(z_{ve})_{dim}, z_{vB}$	ωD^2 (area rate)	$\omega^2 D^2$ (velocity) ²	$1 - \mu$
6	μ_{ve}	M_x, x_{ve}	M_y, y_{ve}	M_z, z_{ve}	$\omega^2 D^3$ (mass)	$\omega^3 D^3$ (mass rate)	$\kappa, \Delta \tau$
7	$(\dot{\mu}_{ve})_{dim}, \dot{\mu}_{vB}$	$(\dot{x}_{ve})_{dim}, \dot{x}_{vB}$	$(\dot{y}_{ve})_{dim}, \dot{y}_{vB}$	$(\dot{z}_{ve})_{dim}, \dot{z}_{vB}$	Δt_k	Δt	$\mu_{vaverage}$
8	$\dot{\mu}_{ve}$	M_x, \dot{x}_{ve}	M_y, \dot{y}_{ve}	M_z, \dot{z}_{ve}	$(\Delta t)^2$	$\ddot{\mu}_{vaverage}$	$D, -M$
9	r_{vsB}	x_{vsB}	y_{vsB}	z_{vsB}	$x_{vsB}, (\sigma_{vs_e})_x$	$y_{vsB}, (\sigma_{vs_e})_y$	$z_{vsB}, (\sigma_{vs_e})_z$
10	r_{vs_e}	x_{vs_e}	y_{vs_e}	z_{vs_e}	$\ddot{x}_{vavg}, \ddot{x}_{vavg}$	$\ddot{y}_{vavg}, \ddot{y}_{vavg}$	$\ddot{z}_{vavg}, \ddot{z}_{vavg}$
11	v_{vsB}	\dot{x}_{vsB}	\dot{y}_{vsB}	\dot{z}_{vsB}	\dot{x}_{vsB}, n_x	\dot{y}_{vsB}, n_y	\dot{z}_{vsB}, n_z
12		$\dot{x}_{vs_e}, e_x + \frac{x_{vs_e}}{r_{vs_e}}$	$\dot{y}_{vs_e}, e_y + \frac{y_{vs_e}}{r_{vs_e}}$	$\dot{z}_{vs_e}, e_z + \frac{z_{vs_e}}{r_{vs_e}}$	$M_s, e_x + \frac{x_{vs_e}}{r_{vs_e}}$	$M_s, e_y + \frac{y_{vs_e}}{r_{vs_e}}$	$e_z + \frac{z_{vs_e}}{r_{vs_e}}$
13			t_p	Δt_{MA}	$1 - e_e^2$	$(1 - e_e^2)^{1/2}$	$\frac{k^2}{\mu_{vavg}}$
14	e_e	e_{x_e}	e_{y_e}	e_{z_e}	e_e^2, e_{x_e}	$\cos(t_e), e_{y_e}$	$\sin(t_e), e_{z_e}$
15	$(k)_{dim}$	$(k_x)_{dim}$	$(k_y)_{dim}$	$(k_z)_{dim}$	b_B, X_B	E_B	$a - r_{vsB}, \text{ or } \vec{r}_{vsB} \cdot \vec{r}_{vsB}$
16	k	k_x	k_y	k_z	k_x, k_e^2, b_e, X_e	k_y, E_e	$k_z, a - r_{vs_e}, \text{ or } \vec{r}_{vs_e} \cdot \vec{r}_{vsB}$

Table 5.8 W-Array -- Virtual Mass Propagation Variables

6. INDIVIDUAL SUBROUTINE DOCUMENTATION

This chapter contains the individual documentation for all the subroutines in the STEAP II series. The following information is given for each subroutine.

1. Purpose: The tasks performed by the subroutine.
2. Calling Sequence: The statement by which the subroutine is called.
3. Arguments: The arguments in the calling sequence, their definition, and identification as input, output, or both.
4. Subroutines Supported: A list of subroutines calling the subroutine being documented.
5. Subroutines Required: A list of subroutines called by the subroutines being documented.
6. Local Symbols: The internal (non-common) variables used in the subroutine and their definitions.
7. Common Computed/Used: A list of variables appearing in common blocks which are both computed and used (see Chapter 3 for definitions).
8. Common Computed: A list of common variables which are set in the program.
9. Common Used: A list of common variables only used by the subroutine.
10. Analysis: The detailed mathematical analysis on which the subroutine is based (if applicable).
11. Flowchart: A flowchart of the operation of the program (if required).

The reader is referred to Chapter 4 for an index of all subroutines of STEAP II (Tables 4.1 and 4.2) and for the calling hierarchies of the basic subprograms of STEAP II (Figures 4.1 to 4.4).

SUBROUTINE ATANH

PURPOSE: TO FIND THE ANGLE Y WHOSE TANH IS X

CALLING SEQUENCE: CALL ATANH(X,Y)

ARGUMENTS: X I TANH(Y)

Y O ANGLE DESIRED

SUBROUTINES SUPPORTED: BATCON

LOCAL SYMBOLS: T1 INTERMEDIATE VARIABLE

T2 INTERMEDIATE VARIABLE

T3 INTERMEDIATE VARIABLE

SUBROUTINE ATCEGV

PURPOSE: TO COMPUTE EIGENVALUES AND EIGENVECTORS OF ACTUAL TARGET
CONDITION 2ND MOMENT MATRIX

CALLING SEQUENCE: CALL ATCEGV(III,ATC,EDT,FOV)

ARGUMENTS: III I NUMBER OF ROWS IN ATC MATRIX

ATC I ACTUAL TARGET CONDITION MATRIX

EDT I ACTUAL TARGET STATE DEVIATION MEANS

FOV I FINAL OFF-DIAGONAL ANNIHILATION TERM FOR JACOBI

SUBROUTINES SUPPORTED: GENGID

SUBROUTINES REQUIRED: HYELS JACOBI

COMMON USED: IHYP1

LOCAL SYMBOLS: DUM1 OUTPUT MATRIX FOR JACOBI

EGVL EIGENVALUES

PEIG INTERMEDIATE ARRAY

ROW INTERMEDIATE VECTOR

S ATC COVARIANCE ARRAY(3,3)

SDUM ATC COVARIANCE ARRAY(2,2) FOR JACOBI

SSDUM ATC COVARIANCE ARRAY(2,2) FOR HYELS

SUBROUTINE BATCON

PURPOSE: TO FIND POSITION, VELOCITY, TRUE ANOMALY, RADIUS AND VELOCITY MAGNITUDE AT TIME T GIVEN POSITION AND VELOCITY AT TIME 0.

CALLING SEQUENCE: CALL BATCON(U,R,V,T,RT,VT)

ARGUMENTS: U I GRAVITATIONAL CONSTANT
 R I POSITION VECTOR
 V I VELOCITY VECTOR
 T I TIME AT WHICH POSITION AND VELOCITY IS DESIRED
 RT O POSITION AT TIME T
 VT O VELOCITY AT TIME T

SUBROUTINES SUPPORTED: BEPS

SUBROUTINES REQUIRED: ATANH ZRANS

LOCAL SYMBOLS: A SEMIMAJOR AXIS OF THE CONIC
 ACC ACCURACY LEVEL____
 AMOM INTERMEDIATE VECTOR
 ANS INTERMEDIATE ARGUMENT
 ARG INTERMEDIATE ARGUMENT
 A0 RECIPROCAL OF A
 C COS OR COSH OF X
 CF0 COSINE TRUE ANOMALY
 CF1 INTERMEDIATE VARIABLE
 C1 INTERMEDIATE VARIABLE
 E ECCENTRICITY
 EPS INTERMEDIATE VARIABLE
 E0 INTERMEDIATE VARIABLE
 E1 INTERMEDIATE VARIABLE
 FINF INTERMEDIATE VARIABLE

FT	TRUE ANOMALY
F0	INTERMEDIATE VARIABLE
F1	INTERMEDIATE VARIABLE
F2	INTERMEDIATE VARIABLE
H0	INTERMEDIATE VARIABLE
H1	INTERMEDIATE VARIABLE
K	COUNTER
LM	INTERMEDIATE VARIABLE
P	SEMI-LATUS RECTUM OF THE CONIC
PI	MATHEMATICAL CONSTANT
RD	INTERMEDIATE VARIABLE
RDT	INTERMEDIATE VARIABLE
RM	MAGNITUDE OF POSITION VECTOR AT TIME 0
RMT	MAGNITUDE OF POSITION VECTOR AT TIME T
RRDT	INTERMEDIATE VARIABLE
S	SIN OR SINH OF X
SF0	SIN TRUE ANOMALY
SF1	INTERMEDIATE VARIABLE
T1	INTERMEDIATE VARIABLE
T2	INTERMEDIATE VARIABLE
T3	INTERMEDIATE VARIABLE
UDTDX	INTERMEDIATE VARIABLE
UTXN	INTERMEDIATE VARIABLE
VM	MAGNITUDE OF VELOCITY VECTOR AT TIME 0
VMT	MAGNITUDE OF VELOCITY VECTOR AT TIME T
X	INTERMEDIATE VARIABLE

BATCON Analysis

BATCON is a conic propagator using the Battin universal variable formulation. A total derivation is too involved to be given here; rather the results of Battin's work will be given here.

Let the initial state of a point mass moving under the influence of a gravitational force μ be given by \vec{r}_0, \vec{v}_0 . It is required to determine the state \vec{r}, \vec{v} at a time T units later. It is useful to introduce the parameters

$$\begin{aligned}\sigma_0 &= \frac{\vec{r}_0 \cdot \vec{v}_0}{\sqrt{\mu}} \\ \alpha &= \frac{2}{r_0} - \frac{v_0^2}{\mu}\end{aligned}\quad (1)$$

Battin's approach is to introduce a new independent variable $x(t)$ in place of time by the relation

$$\frac{dx}{dt} = \frac{\sqrt{\mu}}{r(t)} \quad x(0) = 0 \quad (2)$$

This parametrization greatly simplifies the conic propagation problem. For suppose that the value of x corresponding to $t = T$ is given by X , i.e. $x(T) = X$. Then the final state is given by

$$\begin{aligned}\vec{r} &= R_1(X) \vec{r}_0 + R_2(X) \vec{v}_0 \\ \vec{v} &= V_1(X) \vec{r}_0 + V_2(X) \vec{v}_0\end{aligned}\quad (3)$$

where

$$\begin{aligned}R_1(X) &= 1 - \frac{1}{r_0} U_2(X) & R_2(X) &= \frac{1}{\sqrt{\mu}} \left[r_0 U_1(X) + \sigma_0 U_2(X) \right] \\ V_1(X) &= -\frac{\sqrt{\mu}}{r_0 \dot{r}_0} U_1(X) & V_2(X) &= 1 - \frac{1}{r_0} U_2(X)\end{aligned}\quad (4)$$

and where

$$\begin{aligned}
 U_0(X) &= \begin{cases} \cos \alpha X & \alpha > 0 \\ \cosh \sqrt{-\alpha} X & \alpha < 0 \end{cases} & U_1(X) &= \begin{cases} \frac{\sin \sqrt{\alpha} X}{\sqrt{\alpha}} & \alpha > 0 \\ \frac{\sinh \sqrt{-\alpha} X}{\sqrt{-\alpha}} & \alpha < 0 \end{cases} \\
 U_2(X) &= \frac{1 - U_0(X)}{\alpha} & U_3(X) &= \frac{X - U_1(X)}{\alpha}
 \end{aligned} \quad (5)$$

The problem is thus reduced to the determination of X . X is generated iteratively by the recursive formulae

$$x_{n+1} = x_n - \frac{\sqrt{\mu} t_n - \sqrt{\mu} t}{r_n} = x_n - \Delta x \quad (6)$$

where

$$\begin{aligned}
 \sqrt{\mu} t_n &= r_0 U_1(x'_n) + \sigma U_2(x'_n) + U_3(x_n) \\
 r_n &= r_0 U_0(x'_n) + \sigma U_1(x'_n) + U_2(x_n)
 \end{aligned} \quad (7)$$

To start the process the initial guess is set to

$$x'_0 = \frac{\sqrt{\mu} T}{r_0} \left\{ 1 - \frac{\sigma_0}{2r_0^2} \sqrt{\mu} T + \frac{1}{6r_0^4} \left[3\sigma_0^2 - r_0(1 - \alpha r_0) \right] \mu T^2 \right\} \quad (8)$$

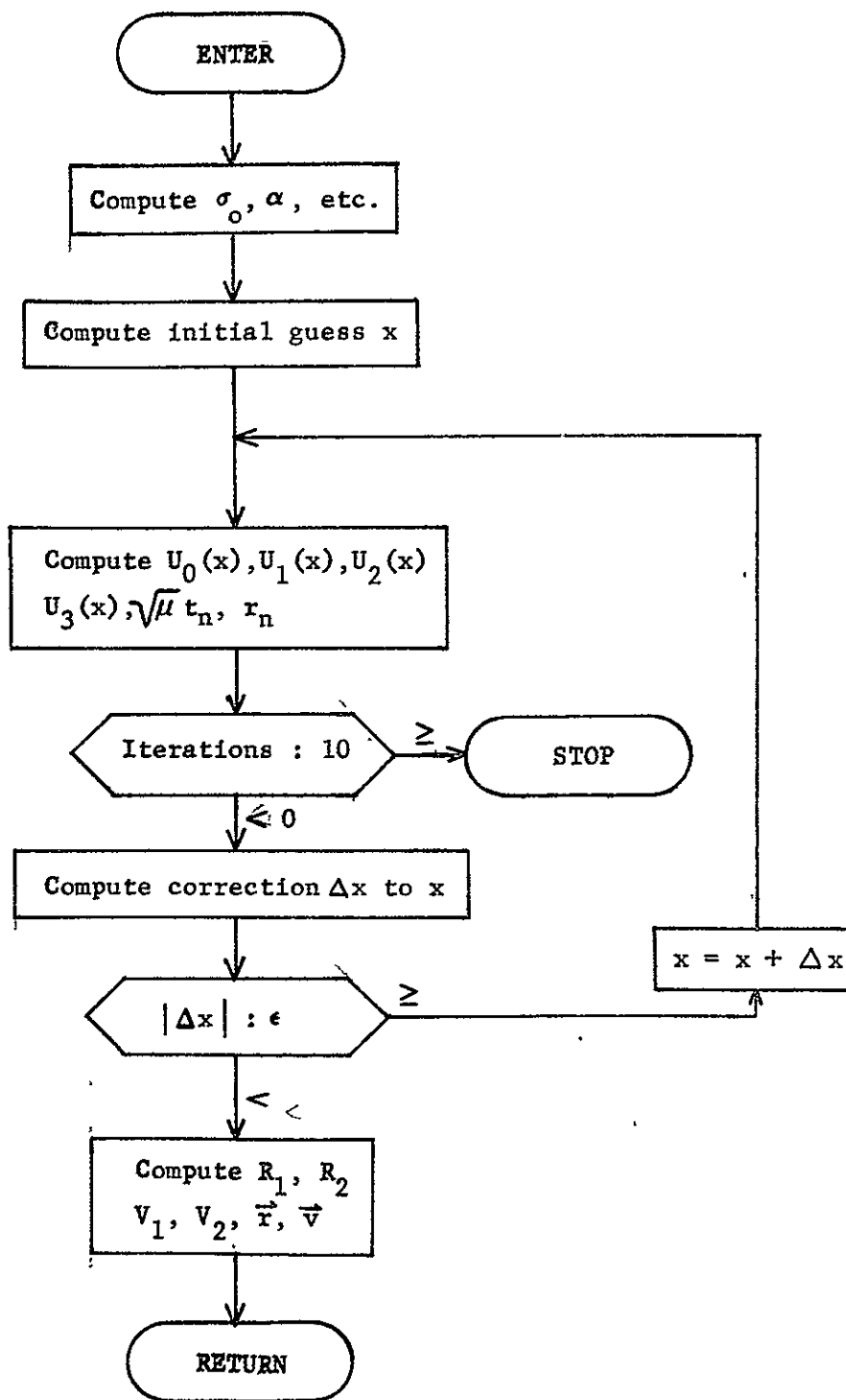
The program sets $X = x_n$ when the correction Δx is less than 10^{-8} . It terminates if the number of iterations exceeds 10.

References:

Battin, R.H., Astronautical Guidance, McGraw-Hill Book Co., New York, 1964.

Battin, R. H. and Fraser, D.C., Space Guidance and Navigation, AIAA Professional Study Series, 1970.

BATCON Flowchart



SUBROUTINE BEPS

PURPOSE: TO COMPUTE POSITION/VELOCITY COVARIANCE AT THE B-PLANE
AND TO COMPUTE THE STATE TRANSITION MATRIX RELATING
CHANGE IN POSITION/VELOCITY TO B-PLANE PARAMETER CHANGES

CALLING SEQUENCE: CALL BEPS(GMU,R,V,GAMMA,BPS)

ARGUMENTS: GMU I GRAVITATIONAL CONSTANT OF TARGET PLANET
R I POSITION VECTOR AT TIME OF EVENT
V I VELOCITY VECTOR AT TIME OF EVENT
GAMMA O STATE TRANSITION MATRIX RELATING STATE TO
B-PLANE PARAMETERS
BPS O POSITION/VELOCITY COVARIANCE AT B-PLANE

SUBROUTINES SUPPORTED: PRED

SUBROUTINES REQUIRED: CAREL BATCOX CONC2 BPLANE

LOCAL SYMBOLS: A SEMIMAJOR AXIS OF THE CONIC
BPV B-PLANE VARIABLES VECTOR
CF INTERMEDIATE VARIABLE
COSFX INTERMEDIATE VARIABLE
DELT TIME DIFFERENCES BETWEEN SOI AND PERIAPSIS
E ECCENTRICITY OF THE CONIC
FX INTERMEDIATE VARIABLE
HSINCF INTERMEDIATE VARIABLE
PP DUMMY ARRAY FOR CAREL CALL
QQ DUMMY ARRAY FOR CAREL CALL
WW DUMMY ARRAY FOR CAREL CALL
PSI STATE TRANSITION MATRIX
PVALUE POSITION/VELOCITY PERTURBATION VECTOR
RN DUMMY VECTOR FOR BPLANE CALL
SN DUMMY VECTOR FOR BPLANE CALL

TN	DUMMY VECTOR FOR BPLANE CALL
RR	INTERMEDIATE VECTOR
VV	INTERMEDIATE VECTOR
RS	POSITION AT THE B-PLANE
VS	VELOCITY AT THE B-PLANE
RSI	SPHERE OF INFLUENCE RADIUS IN KM.
SINFX	INTERMEDIATE VARIABLE
SUM	INTERMEDIATE VARIABLE
TA	INSTANTANEOUS TRUE ANAMOLY OF THE CONIC
TANCF	INTERMEDIATE VARIABLE
TFP	TIME FROM PERIAPSIS
TPSI	INTERMEDIATE VARIABLE
W	ARGUMENT OF PERIAPSIS OF THE CONIC
XI	INCLINATION OF THE CONIC TO REF. FRAME
XN	LONGITUDE OF ASCENDING NODE OF THE CONIC

COMMON USED: ALNGTH NTP SPHERE

SUBROUTINE BIAIM

PURPOSE: TO PERFORM BIASED AIMPOINT GUIDANCE.

CALLING SEQUENCE: CALL BIAIM(RI,TEVN)

ARGUMENTS: RI I NOMINAL SPACECRAFT STATE AT TIME OF BIASED
AIMPOINT GUIDANCE EVENT

TEVN I TIME OF BIASED AIMPOINT GUIDANCE EVENT

SUBROUTINES SUPPORTED: GUISHM GUIDM

SUBROUTINES REQUIRED: MATIN POICOM PSIM QCOMP

LOCAL SYMBOLS

ADA1	VARIATION MATRIX AT TIME T(J+1)
BB	RIGHT HALF PARTITION OF ADA1 MATRIX
CSQ	CONSTANT DEFINING CONSTRAINT ELLIPSE
C1	A COEFFICIENT IN THE NECESSARY CONDITION
C2	A COEFFICIENT IN THE NECESSARY CONDITION
C3	A COEFFICIENT IN THE NECESSARY CONDITION
C4	A COEFFICIENT IN THE NECESSARY CONDITION
C5	A COEFFICIENT IN THE NECESSARY CONDITION
C	SQUARE ROOT OF CSQ
DELMU	AIMPOINT BIAS IN IMPACT PLANE
DENOM	INTERMEDIATE VARIABLE
DET	DETERMINANT OF PROJECTED TARGET CONDITION COVARIANCE MATRIX
DVBIAS	BIAS VELOCITY CORRECTION
DVTT	TOTAL VELOCITY CORRECTION IF BIAS IS REMOVED
DVT	TOTAL VELOCITY CORRECTION IF BIAS IS APPLIED
DVUPP	UPDATE VELOCITY ITERATE
D1	PARTIAL DERIVATIVE USED IN NEWTON ITERATION TECHNIQUE

D2	PARTIAL DERIVATIVE USED IN NEWTON ITERATION TECHNIQUE
D3	PARTIAL DERIVATIVE USED IN NEWTON ITERATION TECHNIQUE
D4	PARTIAL DERIVATIVE USED IN NEWTON ITERATION TECHNIQUE
EE	MATRIX DEFINING FUNCTION TO BE MINIMIZED
IKNT	COUNTER ON NEWTON ITERATION LOOP
IS	INDEX OF NEXT GUIDANCE EVENT
ITRN	COUNTER ON OUTER ITERATION LOOP
NDIM1S	STORAGE FOR NDIM1
NDIM2S	STORAGE FOR NDIM2
PHII	INVERSE OF STATE TRANSITION MATRIX
PHI1	INTERMEDIATE ARRAY
PSIJ1	GUIDANCE MATRIX PSI AT T(J+1)
PSIJ	GUIDANCE MATRIX PSI AT EVENT TIME T(J)
QUOT	INTERMEDIATE VARIABLE
RF	DUMMY VECTOR
SAVET	STORAGE FOR TRTM1
SUM1	INTERMEDIATE VARIABLE
SUM	INTERMEDIATE VARIABLE
TWOE	CONSTANT DEFINING CONSTRAINT ELLIPSE
VCA	SPACECRAFT CLOSEST APPROACH VELOCITY RELATIVE TO TARGET PLANET
XK4	INTERMEDIATE VARIABLE
XMU	MOST RECENT IMPACT PLANE AIMPOINT
XM1	AIMPOINT ITERATE
XM2	AIMPOINT ITERATE
XM	STORAGE FOR MOST RECENT AIMPOINT ITERATE

XN1 NEGATIVE OF CONSTRAINT EQUATION EVALUATED
 AT MOST RECENT AIMPOINT ITERATE

 XN2 NEGATIVE OF NECESSARY CONDITION EVALUATED
 AT MOST RECENT AIMPOINT ITERATE

 YY INTERMEDIATE VARIABLE

 ZH AIMPOINT INCREMENT FOR MOST RECENT
 ITERATION

 ZK AIMPOINT INCREMENT FOR MOST RECENT
 ITERATION

COMMON COMPUTED/USED:

A	CR	DVN	DVRB	DVUP
EXEC	IBIAS	IIGP	PHI2	RCA
TMPR	TRTM1	XMUS		

COMMON COMPUTED:

DELTM

COMMON USED:

ADA	ALNGTH	ATRANS	DUMMYQ	EN3
IDENS	IEND	IGUID	II	ISTMC
ITR	NTP	ONE	PHI	PMASS
POI	PROBI	RADIUS	TINJ	TM
TNOMB	TNOMC	TWO	T3	VINF
XLAMI	XLAM	ZERO		

BIAIM Analysis

Subroutine BIAIM performs biased aimpoint guidance computations. If planetary quarantine constraints are in effect at injection or at a midcourse correction, and if the nominal aimpoint does not satisfy these constraints, subroutine BIAIM will compute a biased aimpoint and the required bias velocity correction such that the constraints are satisfied and some performance functional is minimized.

Aimpoint biasing is performed in the impact plane and as such permits only two degrees of freedom in the selection of the biased aimpoint. The general aimpoint in the impact plane will be denoted by the 2-dimensional vector $\vec{\mu}_j$, where the j-subscript indicates that the biased aimpoint guidance event is occurring at time t_j . Three midcourse guidance policies are available in STEAP, and it will be necessary to relate $\vec{\mu}_j$ to the specific aimpoint for each of these three policies. These relationships are summarized below:

(a) Two-variable B-plane (2VBP):

$$\vec{\mu}_j = \begin{bmatrix} B \cdot T \\ B \cdot R \end{bmatrix} \quad (1)$$

(b) Three-variable B-plane (3VBP):

$$\vec{\mu}_j = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} B \cdot T \\ B \cdot R \\ t_{SI} \end{bmatrix} \quad (2)$$

(c) Fixed-time-of-arrival (FTA):

$$\vec{\mu}_j = A \vec{r}_{CA} \quad (3)$$

where \vec{r}_{CA} is the nominal closest approach position of the spacecraft relative to the target planet. Coordinate transformation A projects the 3-dimensional vector \vec{r}_{CA} (referred to ecliptic coordinates) into an equivalent FTA impact plane which is defined to be the plane containing \vec{r}_{CA} and perpendicular to the spacecraft closest approach velocity \vec{v}_{CA} relative to the target planet. If the ecliptic coordinates of \vec{r}_{CA} and \vec{v}_{CA} are denoted by r_x, r_y, r_z and v_x, v_y, v_z , respectively, then the transformation A is given by

$$A = \begin{bmatrix} \frac{r_x}{r_{CA}} & \frac{r_y}{r_{CA}} & \frac{r_z}{r_{CA}} \\ \frac{r_y v_z - r_z v_y}{r_{CA} v_{CA}} & \frac{r_z v_x - r_x v_z}{r_{CA} v_{CA}} & \frac{r_x v_y - r_y v_x}{r_{CA} v_{CA}} \end{bmatrix} \quad (4)$$

Spacecraft state variations at t_j are related to aimpoint variations (target condition variations) by the variation matrix η_j , which is always available prior to calling BIAIM. Thus, the statistical state dispersions about the nominal following the guidance correction at t_j and represented by the control covariance $P_{c_j}^+$, can be related to the dispersions about the nominal aimpoint represented by W_j^+ according to the equation

$$W_j^+ = \eta_j P_{c_j}^+ \eta_j^T \quad (5)$$

The control covariance $P_{c_j}^+$ is computed from

$$P_{c_j}^+ = P_{k_j}^- + \begin{bmatrix} 0 & 1 & 0 \\ -0 & -1 & -0 \\ 0 & 1 & \tilde{Q}_j \end{bmatrix} \quad (6)$$

where $P_{k_j}^-$ is the knowledge covariance prior to the guidance event and \tilde{Q}_j is the execution error covariance.

Transformations employed in equations (1) through (3) can also be employed to project W_j^+ into the impact plane. The resulting projection is denoted by the covariance \mathcal{A}_j , and is obtained from W_j^+ according to the following equations:

$$(a) \quad 2VBP : \quad \mathcal{A}_j = W_j^+ \quad (7)$$

$$(b) \quad 3VBP : \quad \mathcal{A}_j = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} W_j^+ \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \quad (8)$$

$$(c) \quad FTA : \quad \mathcal{A}_j = A W_j^+ A^T \quad (9)$$

With covariance \mathcal{L}_j available, it is now possible to compute the probability of impact PØI. Assuming the probability density function associated with \mathcal{L}_j is gaussian and nearly constant over the target planet capture area permits us to compute PØI using the equation

$$PØI = \pi R_c^2 p \quad (10)$$

where R_c is the target planet capture radius and p represents the gaussian density function evaluated at the target planet center and is given by

$$p = \frac{1}{2\pi |\mathcal{L}_j|^{\frac{1}{2}}} \exp \left[-\frac{1}{2} \vec{\mu}^{*T} \mathcal{L}_j^{-1} \vec{\mu}^* \right] \quad (11)$$

The nominal impact plane aimpoint is denoted by $\vec{\mu}^*$. Subroutine BIAIM calls subroutine PØICØM to perform the computations involved in equations (7) through (11).

Capture radius R_c is simply the physical radius R_p of the target planet if the FTA guidance policy is employed, while for the two B-plane policies the capture radius is given by

$$R_c = R_p \sqrt{1 + \frac{2\mu_p}{V_\infty^2 R_p}} \quad (12)$$

where μ_p is the target planet gravitational constant and V_∞ is the hyperbolic excess velocity.

If the probability of impact PØI does not exceed the permissible impact probability P_I , and if the nominal aimpoint has not been previously biased, we simply return to subroutine GUIDM (or GUIDSIM). If the nominal aimpoint has been previously biased, a velocity correction $\Delta \vec{V}_{RB_j}$ required to remove that bias is computed prior to returning. But if PØI exceeds P_I , an aimpoint bias $\delta \vec{\mu}_j$ and the associated bias velocity correction $\Delta \vec{V}_{B_j}$ must be computed. Before describing the details of the biasing technique it is necessary to define the relationship between $\Delta \vec{V}_j$ and $\delta \vec{\mu}_j$ for linear midcourse guidance policies.

Linear impulsive guidance policies have form

$$\Delta \vec{V}_j = \Gamma_j \delta \vec{X}_j \quad (13)$$

where Γ_j is the guidance matrix and $\delta\vec{X}_j$ is the spacecraft state deviation from the targeted nominal trajectory. (These guidance policies are discussed in more detail in the subroutine GUIIS analysis section.) Such guidance policies can be readily generalized to account for changes in the target conditions from their nominal values. This generalized version of equation (13) has form

$$\Delta\vec{V}_j = \Gamma_j \delta\vec{X}_j + \Psi_j \delta\vec{\mu}_j \quad (14)$$

where Ψ_j can also be referred to as a guidance matrix. For the purposes of the BIAIM analysis, we shall assume that $\delta\vec{\mu}_j$ in equation (14) is always an aimpoint change in the impact plane. Thus, Ψ_j will be a 3x2 guidance matrix. The derivation of the Ψ_j matrix is quite similar to the derivation of the Γ_j matrix and will not be presented here. If we partition the previously discussed variation matrix η_j as follows:

$$\eta_j = \begin{bmatrix} \eta_1 & \eta_2 \end{bmatrix} \quad (15)$$

then the Ψ_j matrices for the three midcourse guidance policies are given by the following equations:

$$(a) \quad 2VBP : \quad \Psi_j = \eta_2^{-T} (\eta_2 \eta_2^T)^{-1} \quad (16)$$

$$(b) \quad 3VBP : \quad \Psi_j = \eta_2^{-1} \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \quad (17)$$

$$(c) \quad FTA : \quad \Psi_j = \eta_2^{-1} A^T \quad (18)$$

If an aimpoint bias were to be removed at time t_j , the required velocity correction would be given by

$$\Delta\vec{V}_{RBj} = -\Psi_j \delta\vec{\mu}_j \quad (19)$$

If an aimpoint bias were to be imparted at time t_j , the bias velocity correction would be given by

$$\Delta\vec{V}_{Bj} = \Psi_j \delta\vec{\mu}_j \quad (20)$$

If an aimpoint bias $\delta \vec{\mu}_j^{(1)}$ had been previously imparted, and if a new aimpoint bias $\delta \vec{\mu}_j^{(2)}$ is to be imparted, then the total bias velocity correction would be given by

$$\Delta \vec{v}_{B_j} = \psi_j \left[\delta \vec{\mu}_j^{(2)} - \delta \vec{\mu}_j^{(1)} \right] \quad (21)$$

The general statement of the biased aimpoint guidance problem is as follows: Find an aimpoint $\vec{\mu}_j$ in the impact plane which satisfies the impact probability constraint

$$P\phi I \leq P_I \quad (22)$$

and minimizes a performance functional having form

$$J = (\vec{\mu}_j - \vec{\mu}^*)^T \tilde{A} (\vec{\mu}_j - \vec{\mu}^*) \quad (23)$$

where $\vec{\mu}^*$ is the nominal aimpoint and \tilde{A} is a constant symmetric matrix that will be defined subsequently.

The solution of this problem is detailed in the section on biased aimpoint guidance in the analytical manual. Only the results will be presented here. The assumption of constant probability density over the target planet capture area permits us to rewrite constraint equation (22) as

$$\lambda_1 \mu_1^2 + 2\lambda_3 \mu_1 \mu_2 + \lambda_2 \mu_2^2 = c^2 \quad (24)$$

where

$$c^2 = 2 \ln \left[\frac{R_c^2}{2 |\mathcal{A}|^{\frac{1}{2}} P_I} \right] \quad (25)$$

$$\text{and } \vec{\mu} = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix} \quad \text{and } \mathcal{A}^{-1} = \begin{bmatrix} \lambda_1 & \lambda_3 \\ \lambda_3 & \lambda_2 \end{bmatrix}.$$

The inequality has been replaced by an equality since the solution can be shown to lie on the constraint boundary, which, from inspection of equation (24) is an ellipse centered at the target planet.

If t_j is the time of the final midcourse correction, matrix \tilde{A} will be chosen as a 2x2 identity matrix. The minimization of J is then equivalent to minimization of the miss distance $|\vec{\mu}_j - \vec{\mu}^*|$. If t_j is not the final midcourse correction time, \tilde{A} will be defined as follows:

$$\tilde{A} = \psi_{j+1}^T \psi_{j+1} \quad (26)$$

Here ψ_{j+1} denotes the aimpoint guidance matrix for the next midcourse correction occurring at time t_{j+1} . In this case the minimization of J is equivalent to the minimization of $|\Delta \vec{V}_{RB,j+1}|$, i.e., the velocity required to remove bias $\delta \vec{\mu}_j$ at time t_{j+1} will be minimized. The computation of ψ_{j+1} is based on the variation matrix η_{j+1} , just as ψ_j was based on η_j . However, η_{j+1} can be computed more efficiently by using the relationship

$$\eta_{j+1} = \eta_j \Phi_{j+1,j}^{-1} \quad (27)$$

where $\Phi_{j+1,j}$ is the state transition matrix over $[t_j, t_{j+1}]$.

If we define

$$\tilde{A} = \begin{bmatrix} a_1 & a_3 \\ a_3 & a_2 \end{bmatrix}$$

then the necessary condition for a minimum is given by

$$\begin{aligned} & (a_1 \lambda_3 - a_3 \lambda_1) \mu_1^2 + (a_3 \lambda_2 - a_2 \lambda_3) \mu_2^2 + (a_1 \lambda_2 - a_2 \lambda_1) \mu_1 \mu_2 \\ & + (-a_1 \lambda_3 \mu_1^* - a_3 \lambda_3 \mu_2^* + a_3 \lambda_1 \mu_1^* + a_2 \lambda_1 \mu_2^*) \mu_1 + \\ & (-a_1 \lambda_2 \mu_1^* - a_3 \lambda_2 \mu_2^* + a_3 \lambda_3 \mu_1^* + a_2 \lambda_3 \mu_2^*) \mu_2 = 0 \end{aligned} \quad (28)$$

Thus, our problem is reduced to finding μ_1 and μ_2 which satisfy equations (24) and (28). Since the analytical solution of these equations proved intractable, a standard Newton iteration technique is employed in BIAIM which quickly converges to solutions for μ_1 and μ_2 . The iteration process is started with an initial guess defined as the intersection of the extended $\vec{\mu}^*$ vector and the constraint boundary defined by equation (24). This initial guess is given by

$$\mu_1^o = \begin{pmatrix} \frac{\mu_1^*}{\mu_2^*} \end{pmatrix} \mu_2^o \quad (29)$$

$$\mu_2^o = \operatorname{sgn}(\mu_2^*) \frac{c}{\sqrt{\lambda_1 \left(\frac{\mu_1^*}{\mu_2^*} \right)^2 + 2\lambda_3 \left(\frac{\mu_1^*}{\mu_2^*} \right) + \lambda_2}}$$

where c is defined by equation (25).

In addition to the previously described iteration process, subroutine BIAIM also employs an outer iteration loop which accounts for the dependence of \tilde{Q}_j (equation (6)) on $\delta \tilde{\mu}_j$. The execution error covariance \tilde{Q}_j is a function of the total velocity correction at t_j , but the total velocity correction, in particular $\Delta \tilde{V}_{B_j}$, depends on $\delta \tilde{\mu}_j$. This coupling is resolved by recomputing \tilde{Q}_j at the end of the previously described biasing technique and repeating the biasing cycle until the error function

$$\left| P_{OI} - P_I \right| \leq P_I \times 10^{-3}$$

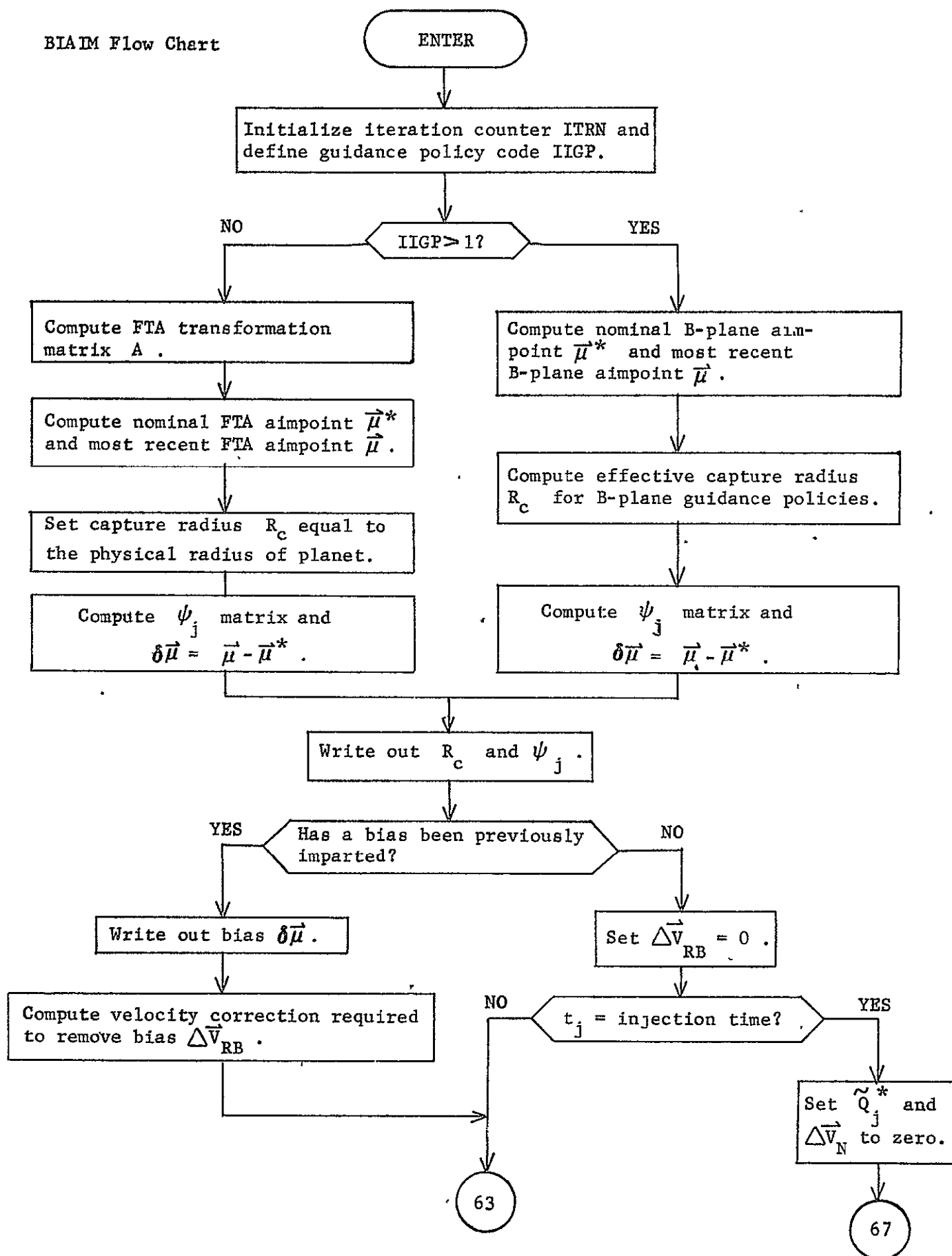
is satisfied. This outer iteration process is not performed, however, if t_j = injection time since at injection equation (6) is replaced by the equation

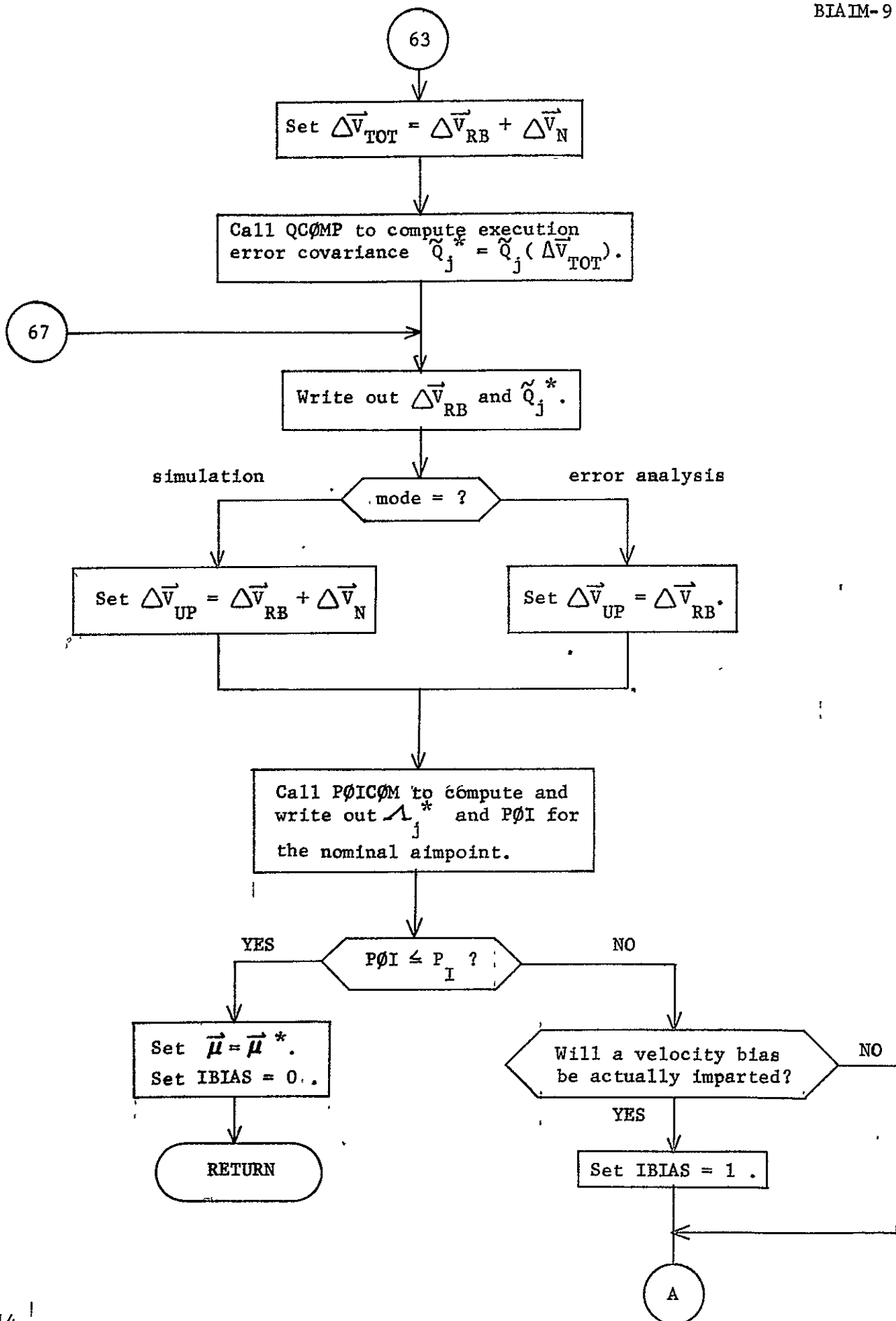
$$P_{c_j} = P_{k_j}$$

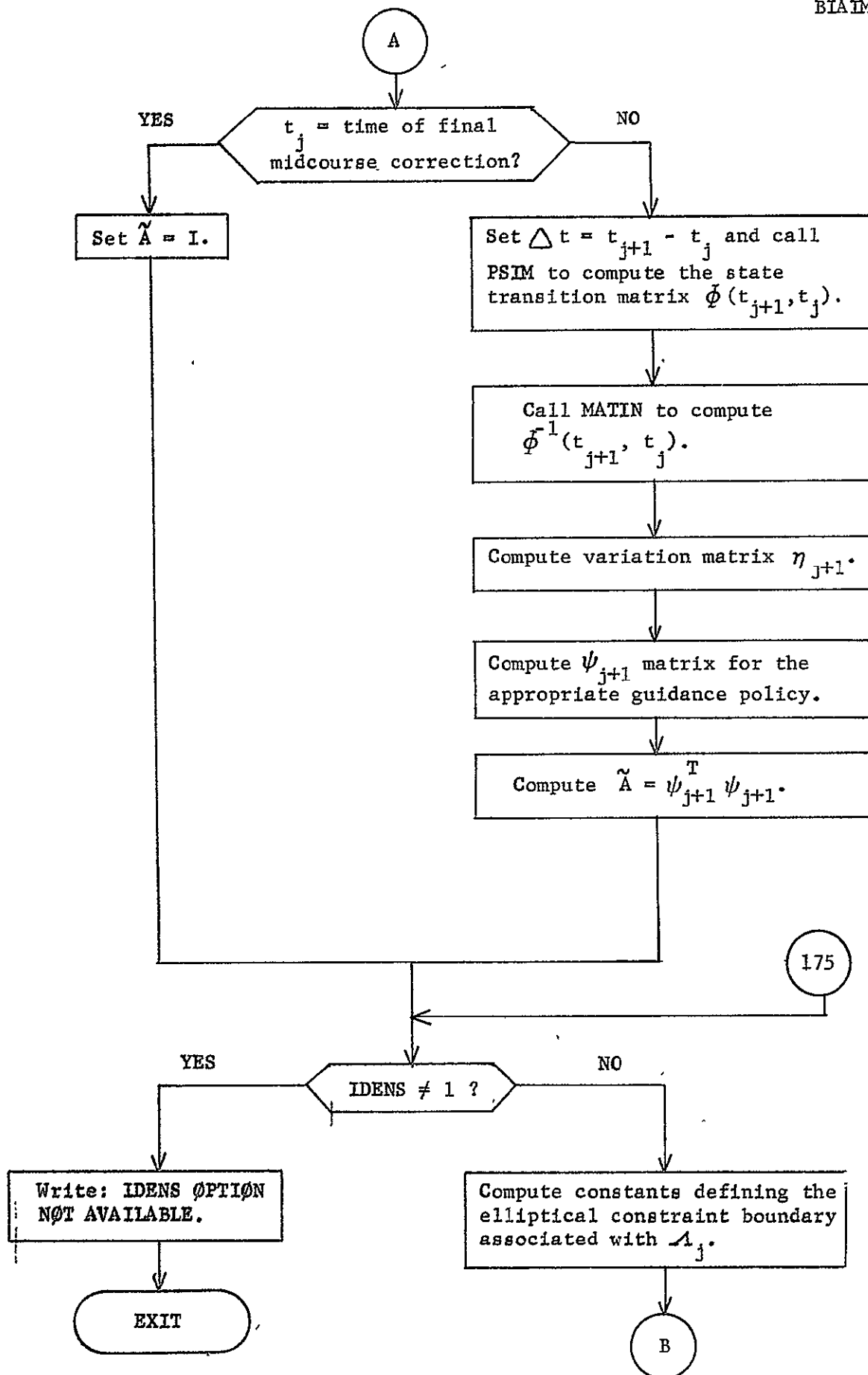
and \tilde{Q}_j is always zero.

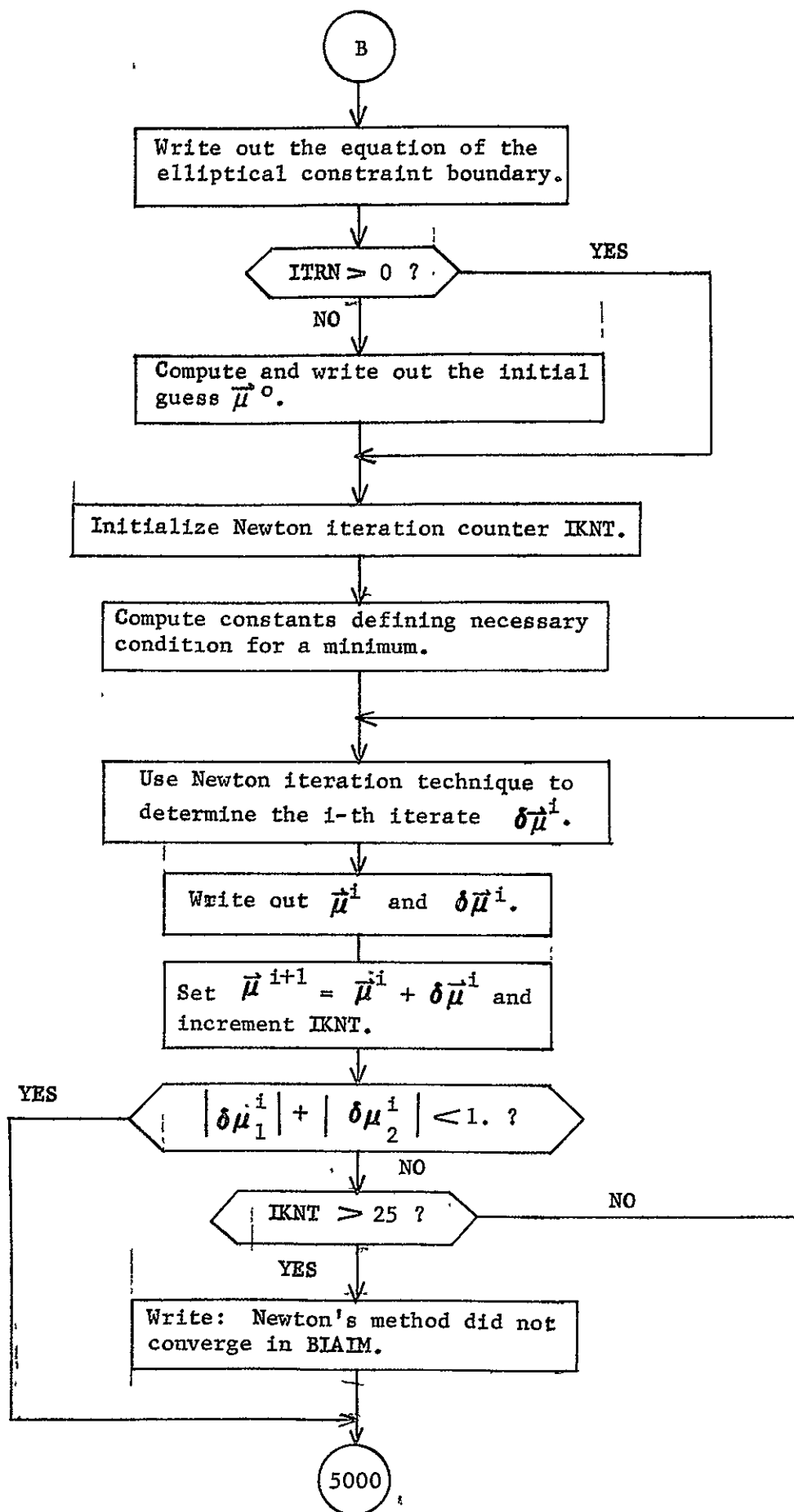
Reference: Mitchell, R. T., and Wong, S. K.: Preliminary Flight Path Analysis Orbit Determination and Maneuver Strategy Mariner Mars 1969. Project Document 138, Jet Propulsion Laboratory, 1968.

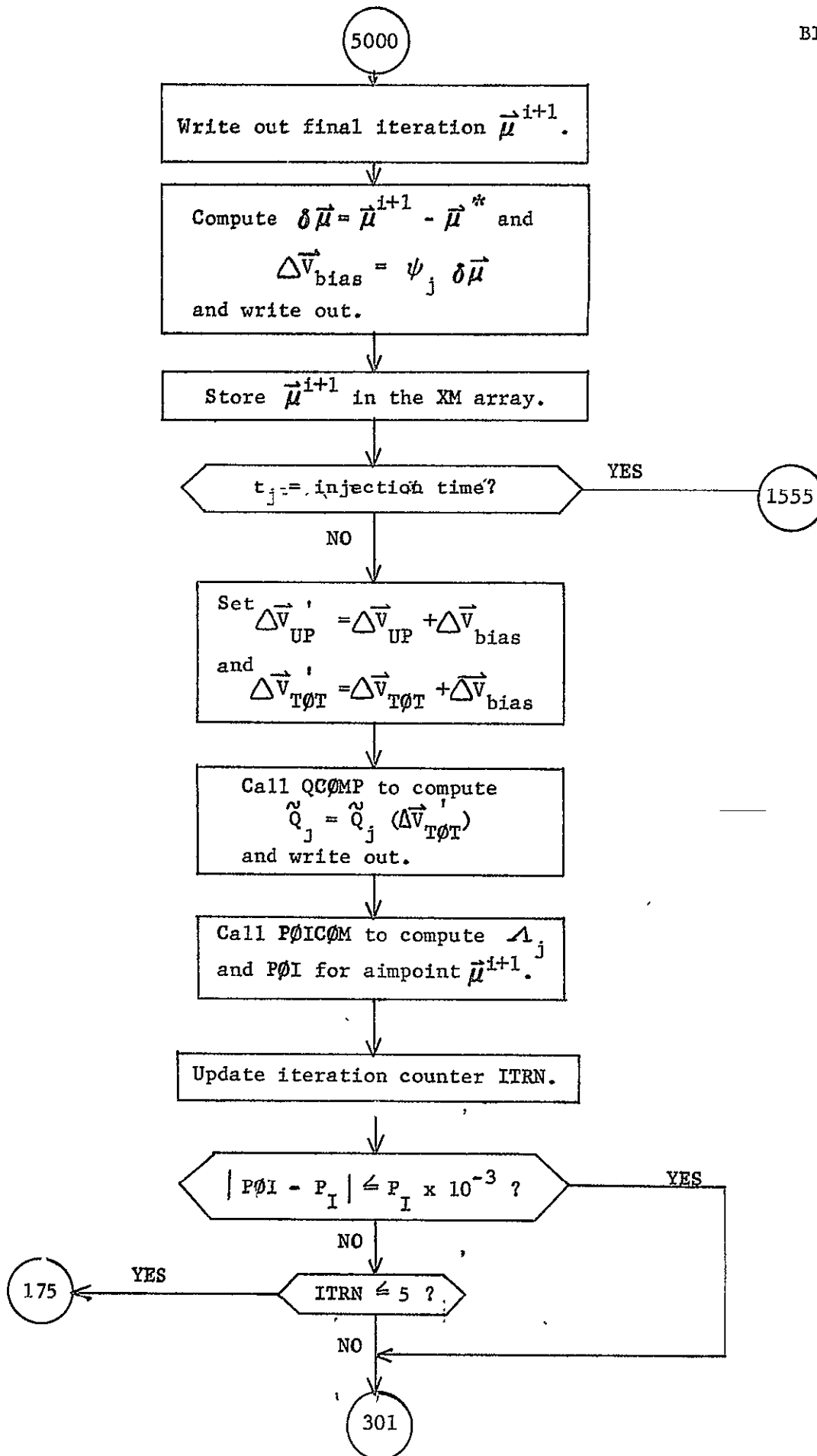
BIAIM Flow Chart

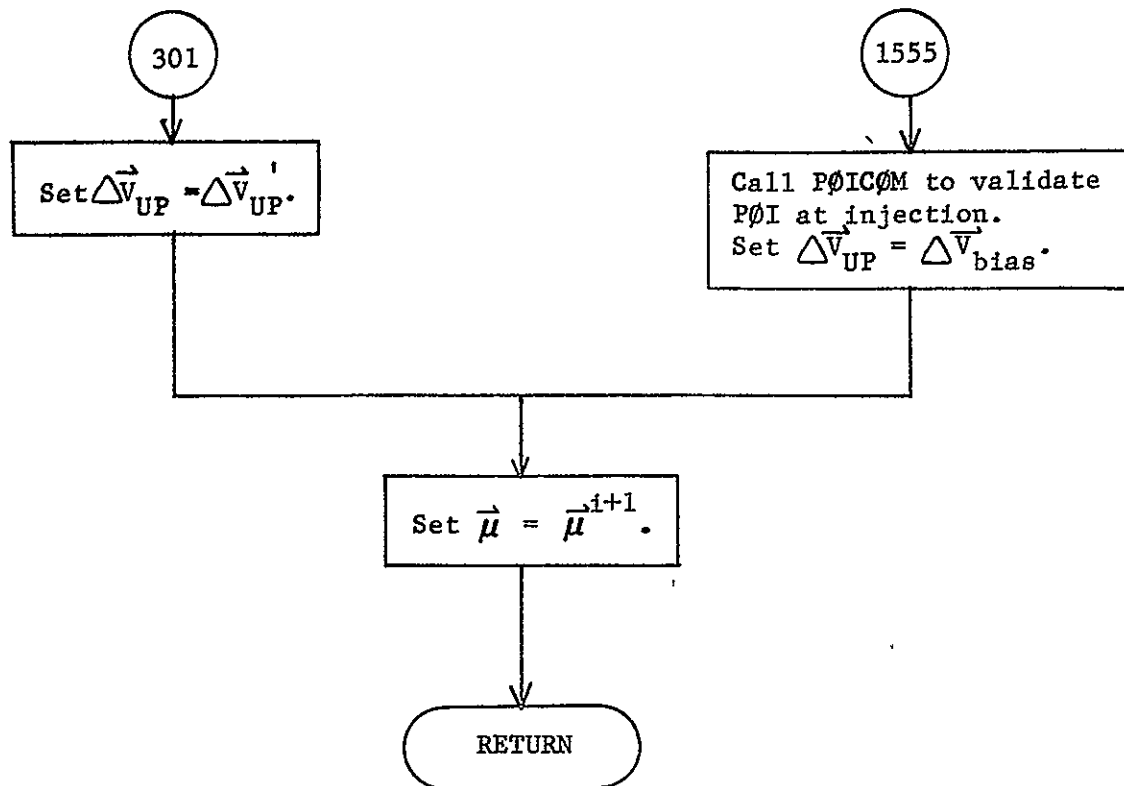












SUBROUTINE BIAS

PURPOSE: COMPUTE THE ACTUAL MEASUREMENT BIAS IN THE SIMULATION PROGRAM

RETURN THE ACTUAL MEASUREMENT BIAS TO BE USED IN THE SIMULATION MODE.

CALLING SEQUENCE: CALL BIAS(MCODE,BVAL)

ARGUMENT: BVAL 0 THE ACTUAL BIAS TO BE USED IN THE MEASUREMENT

MCODE I MEASUREMENT TYPE CODE

SUBROUTINES SUPPORTED: SIMULL

COMMON USED: BIA

BIAS Analysis

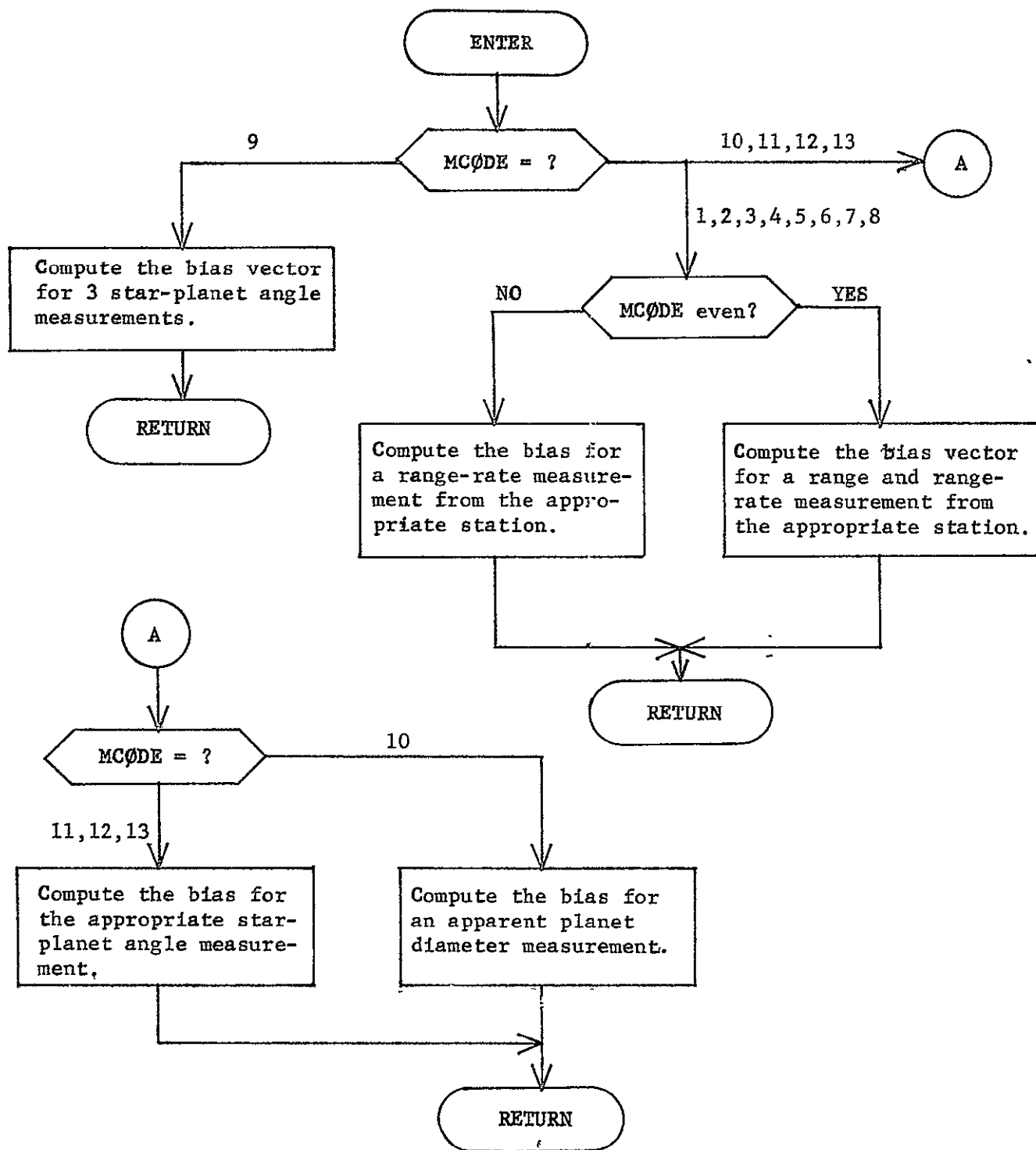
The actual measurement y_k^a at time t_k is given by

$$y_k^a = \underline{y}_k + b_k + \nu_k$$

where \underline{y}_k is the ideal measurement, which would be made in the absence of instrumentation errors, b_k is the actual measurement bias and ν_k represents the actual measurement noise.

The function of subroutine BIAS is to compute the measurement bias b_k for the appropriate measurement type. The constant biases for all measurement devices are stored in the vector BIA. Subroutine BIAS selects the appropriate elements from this vector to construct the actual measurement bias.

BIAS Flow Chart



BLOCK DATA

PURPOSE: TO LOAD CONSTANTS INTO COMMON LOCATIONS USED IN VARIOUS
OTHER PARTS OF THE PROGRAM.

CALLING SEQUENCE: NONE

ARGUMENT: NONE

SUBROUTINES SUPPORTED HALF THE SUBROUTINES USE THE CONSTANTS
STORED BY THIS SUBROUTINE

SUBROUTINES REQUIRED NONE

COMMON LOADED	CN1	CN	ELMNT	EMN	EVNM
	RADIUS	RAD	RMASS	SMJR	SPHERE
	F	MNNAME	PI	PLANET	PMASS
	ST				

BLKDAT Analysis

Subroutine BLKDAT is responsible for setting up constants used in computing ephemeris data for the gravitating bodies.

The arrays set up by BLKDAT and their definitions are as follows:

Array	Definition
CN(80)	Constants defining mean elements for inner planets
ST(50)	Constants defining mean elements for outer planets
SMJR(18)	Constants defining semi-major axes for planets and moon
EMN(15)	Constants defining lunar elements
PMASS(11)	Gravitational constants of sun, planets, and moon
RMASS(11)	Mass of bodies relative to sun
RADIUS(11)	Surface radii of sun, planets, and moon
SPHERE(11)	Sphere of influence radii of sun, planets, and moon
MONTH(12)	Names of months for output purposes
PLANET(11)	Names of planets for output purposes

The definitions of the CN, ST, SMJR, and EMN arrays are provided in Tables 2 through 5 on the following page. The actual constants stored in those arrays are the ephemeris data listed on the next pages following.

The constants stored in the other arrays are given below.

Body	PMASS (AU ³ /day ²)	RMASS*	RADIUS (AU)	SPHERE (AU)
Sun	2.959122083(-4)	1.0	4.66582(-3)	NA
Mercury	4.850(-11)	1.639(-7)	1.617(-5)	7.46(-4)
Venus	7.243(-10)	2.448(-6)	4.044(-5)	4.12(-3)
Earth	8.88757(-10)	3.003(-6)	4.263(-5)	6.18(-3)
Mars	9.5497905(-11)	3.236(-7)	2.279(-5)	3.78(-3)
Jupiter	2.8252(-7)	9.547(-4)	4.7727(-4)	.3216
Saturn	8.454(-8)	2.857(-4)	4.0374(-4)	.3246
Uranus	1.290(-8)	4.359(-5)	1.5761(-4)	.346
Neptune	1.5(-8)	5.069(-5)	1.4906(-4)	.5805
Pluto	7.4(-10)	2.501(-6)	4.679(-5)	.2366
Moon	1.0921748(-11)	3.696(-8)	1.161(-5)	3.71394(-4)

* Truncated from program values.

Array Definitions

Constant	1	Ω	$\tilde{\omega}$	e	M	a	ω	E	a_0	a_1
Mercury	1	2	3	4	5	6	7	8	1	2
Venus	9	10	11	12	13	14	15	16	3	4
Earth	17	18	19	20	21	22	23	24	5	6
Mars	25	26	27	28	29	30	31	32	7	8
Jupiter	33	34	35	36	37	38	39	40	9	10
Saturn	41	42	43	44	45	46	47	48	11	12
Uranus	49	50	51	52	53	54	55	56	13	14
Neptune	57	58	59	60	61	62	63	64	15	16
Pluto	65	66	67	68	69	70	71	72	17	18
Moon	73	74	75	76	77	78	79	80		

Table 1. ELMNT Array -- Conic Elements

Table 2. SMJR Array

Constant	i_0	i_1	i_2	i_3	Ω_0	Ω_1	Ω_2	Ω_3	$\tilde{\omega}_0$	$\tilde{\omega}_1$	$\tilde{\omega}_2$	$\tilde{\omega}_3$	e_0	e_1	e_2	e_3	M_0	M_1	M_2	M_3
Mercury	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Venus	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
Earth	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60
Mars	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80

Table 3. CN Array -- Inner Planet Constants

Constant	i_0	i_1	Ω_0	Ω_1	$\tilde{\omega}_0$	$\tilde{\omega}_1$	e_0	e_1	M_0	M_1
Jupiter	1	2	3	4	5	6	7	8	9	10
Saturn	11	12	13	14	15	16	17	18	19	20
Uranus	21	22	23	24	25	26	27	28	29	30
Neptune	31	32	33	34	35	36	37	38	39	40
Pluto	41	42	43	44	45	46	47	48	49	50

Table 4. ST Array -- Outer Planet Constants

Constant	Ω_0	Ω_1	Ω_2	Ω_3	$\tilde{\omega}_0$	$\tilde{\omega}_1$	$\tilde{\omega}_2$	$\tilde{\omega}_3$	L_0	L_1	L_2	L_3	i	e	a
Moon	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Table 5. EMN Array -- Lunar Constants

Planetary and Lunar Ephemerides

Mean Elements of Mercury

$$i = 0.1222233228 + 3.24776685 \times 10^{-5} T - 3.199770295 \times 10^{-7} T^2$$

$$\Omega = 0.8228518595 + 2.068578774 \times 10^{-2} T + 3.034933644 \times 10^{-6} T^2$$

$$\tilde{\omega} = 1.3246996178 + 2.714840259 \times 10^{-2} T + 5.143873156 \times 10^{-6} T^2$$

$$e = 0.20561421 + 0.00002046 T - 0.000000030 T^2$$

$$M = 1.785111955 + 7.142471000 \times 10^{-2} d + 8.72664626 \times 10^{-9} d^2$$

$$a = 0.3870986 \text{ A.U.} = 57,909,370 \text{ km}$$

Mean Elements of Venus

$$i = 0.0592300268 + 1.7555510339 \times 10^{-5} T - 1.696847884 \times 10^{-8} T^2$$

$$\Omega = 1.3226043500 + 1.570534527 \times 10^{-2} T + 7.155849933 \times 10^{-6} T^2$$

$$\tilde{\omega} = 2.2717874591 + 2.457486613 \times 10^{-2} T + 1.704120089 \times 10^{-5} T^2$$

$$e = 0.00682069 - 0.00004774 T + 0.000000091 T^2$$

$$M = 3.710626172 + 2.796244623 \times 10^{-2} d + 1.682497399 \times 10^{-6} d^2$$

$$a = 0.7233316 \text{ A.U.} = 108,209,322 \text{ km}$$

Mean Elements of Earth

$$i = 0$$

$$\Omega = 0$$

$$\tilde{\omega} = 1.7666368138 + 3.000526417 \times 10^{-2} T + 7.902463002 \times 10^{-6} T^2 \\ + 5.817764173 \times 10^{-8} T^3$$

$$e = 0.01675104 - 0.00004180 T - 0.000000126 T^2$$

$$M = 6.256583781 + 1.720196977 \times 10^{-2} d - 1.954768762 \times 10^{-7} d^2 \\ - 1.22173047 \times 10^{-9} d^3$$

$$a = 1.0000003 \text{ A.U.} = 149,598,530 \text{ km}$$

Mean Elements of Mars

$$i = 0.0322944089 - 1.178097245 \times 10^{-5} T + 2.201054112 \times 10^{-7} T^2$$

$$\Omega = 0.8514840375 + 1.345634309 \times 10^{-2} T - 2.424068406 \times 10^{-8} T^2 \\ - 9.308422677 \times 10^{-8} T^3$$

$$\tilde{\omega} = 5.8332085089 + 3.212729365 \times 10^{-2} T + 2.266503959 \times 10^{-6} T^2 \\ - 2.084698829 \times 10^{-8} T^3$$

$$e = 0.09331290 + 0.000092064 T - 0.000000077 T^2$$

$$M = 5.576840523 + 9.145887726 \times 10^{-3} d + 2.365444735 \times 10^{-7} d^2 \\ + 4.363323130 \times 10^{-10} d^3$$

$$a = 1.5236915 \text{ A.U.} = 227,941,963 \text{ km}$$

Mean Elements of Jupiter

$$i = 0.0228410270 - 9.696273622 \times 10^{-5} T$$

$$\Omega = 1.7355180770 + 1.764479392 \times 10^{-2} T$$

$$\tilde{\omega} = 0.2218561704 + 2.812302353 \times 10^{-2} T$$

$$e = 0.0483376 + 0.00016302 T$$

$$M = 3.93135411 + 1.450191928 \times 10^{-3} d$$

$$a = 5.202803 \text{ A.U.} = 778,331,525 \text{ km}$$

Mean Element of Saturn

$$i = 0.0435037861 - 7.757018898 \times 10^{-8} T$$

$$\Omega = 1.9684445802 + 1.523977870 \times 10^{-2} T$$

$$\tilde{\omega} = 1.5897996653 + 3.419861162 \times 10^{-2} T$$

$$e = 0.0558900 - 0.00034705 T$$

$$M = 3.0426210430 + 5.837120844 \times 10^{-4} d$$

$$a = 9.538843 \text{ A.U.} = 1,426,996,160 \text{ km}$$

Mean Elements of Uranus

$$i = 0.0134865470 + 0.696273622 \times 10^{-6} T$$

$$\Omega = 1.2826407705 + 8.912087493 \times 10^{-3} T$$

$$\tilde{\omega} = 2.9502426085 + 2.834608631 \times 10^{-2} T$$

$$e = 0.0470463 + 0.00027204 \text{ T}$$

$$M = 1.2843599198 + 2.046548840 \times 10^{-4} \text{ d}$$

$$a = (19.182281 - 0.00057008 \text{ T}) \text{ A.U.} = (2,869,640,310 - 85271 \text{ T}) \text{ km}$$

Mean Elements of Neptune

$$i = 0.0310537707 - 1.599885148 \times 10^{-4} \text{ T}$$

$$\Omega = 2.2810642235 + 1.923032859 \times 10^{-2} \text{ T}$$

$$\tilde{\omega} = 0.7638202701 + 1.532704516 \times 10^{-2} \text{ T}$$

$$e = 0.00852849 + 0.00007701 \text{ T}$$

$$M = 0.7204851506 + 1.033089473 \times 10^{-4} \text{ d}$$

$$a = (30.057053 + 0.001210166 \text{ T}) \text{ A.U.} = (4,496,490,000 + 181039 \text{ T}) \text{ km}$$

Mean Elements of Pluto

$$i = 0.2996706970859694$$

$$\Omega = 1.1914337550102258$$

$$\tilde{\omega} = 3.909919302791948$$

$$e = 0.2488033053623924$$

$$M = 3.993890007 + 0.6962635708298997 \times 10^{-4}$$

$$a = 39.37364135300176 \text{ A.U.} = 5,890,213,786,146,730 \text{ km}$$

Mean Elements of Moon

$$i = 5.1453964^{\circ}$$

$$\Omega = 259.183275^{\circ} - 0.0529539222d + 0.002078 \text{ T}^2 + 0.000002 \text{ T}^3$$

$$\tilde{\omega} = 334.329556^{\circ} + 0.1114040803d - 0.010325 \text{ T}^2 - 0.000012 \text{ T}^3$$

$$L = 270.434164^{\circ} + 13.1763965268d - 0.001133 \text{ T}^2 + 0.0000019 \text{ T}^3$$

$$a = .00256954448 \text{ A.U.}$$

$$e = 0.054900489$$

- Note 1: The above elements are referred to the mean equinox and ecliptic of date except for Pluto.
- Note 2: The elements for pluto are oscillating values for epoch 1960 September 23.0 E.T. = J.D. 2437200.5
- Note 3: The time interval from the epoch is denoted by T when measured in Julian centuries of 36,525 ephemeris days, by $D = 3.6525 T$ when measured in units of 10,000 ephemeris days, and by $d = 10,000D = 36,525 T$ when measured in ephemeris days. Times are measured with respect to the epoch 1900 January 0.5 E.T. = J.D. 2415020.0.
- Note 4: Angular relations are expressed in radians for planets and degrees for moon.

-
- References: (1) Space Research Conic Program, Phase III, J.P.L., May 1969
(Planetary constants)
- (2) The American Ephemeris and Nautical Almanac - 1965, U.S. Government Printing Office, Washington, p. 493 (Lunar constants)

SUBROUTINE BPLANE

PURPOSE: TO COMPUTE B-PLANE PARAMETERS

CALLING SEQUENCE: CALL BPLANE(GMX,R,V,BUT,BDR,TF,SDR,SDT,C3,INDX,
RN,SN,TN)

ARGUMENTS: GMX I GRAVITATIONAL CONSTANT OF TARGET PLANET
 R I POSITION VECTOR
 V I VELOCITY VECTOR
 BDT O B DOT T
 BDR O B DOT R
 TF O TIME FROM PERIAPSIS
 SDR O S DOT R
 SDT O S DOT T
 C3 O PLANET DEPARTURE ENERGY
 INDX I =1 FILL RN,SN,TN VECTORS, =2 DO NOT FILL
 RN O R VECTOR
 SN O S VECTOR
 TN O T VECTOR

SUBROUTINES SUPPORTED: BEPS

LOCAL SYMBOLS: A SEMIMAJOR AXIS OF THE CONIC
 AB INTERMEDIATE VARIABLE
 AUXF INTERMEDIATE VARIABLE
 B INTERMEDIATE VARIABLE
 BV B VECTOR
 CTA COSINE TRUE ANOMALY
 C1 ANGULAR MOMENTUM CONSTANT
 E ECCENTRICITY
 NINETY 90.

ONE	1.
P	SEMI-LATUS RECTUM
PI	MATHEMATICAL CONSTANT
PV	P VECTOR
QV	Q VECTOR
RAD	DEGREES PER RADIAN
RD	INTERMEDIATE VARIABLE
RM	INTERMEDIATE VARIABLE
RRD	INTERMEDIATE VARIABLE
RV	R VECTOR
SINHF	INTERMEDIATE VARIABLE
STA	SINE TRUE ANOMALY
SV	S VECTOR
TA	INTERMEDIATE VARIABLE
TANG	INTERMEDIATE VARIABLE
TCA	INTERMEDIATE VARIABLE
TV	T VECTOR
TWO	2.
VX	INTERMEDIATE VARIABLE
WV	INTERMEDIATE VARIABLE
Z	INTERMEDIATE VARIABLE
ZERO	0.

SUBROUTINE CAREL

PURPOSE: TRANSFORM CARTESIAN COORDINATES TO CONIC ELEMENTS

CALLING SEQUENCE: CALL CAREL(GM,R,V,TFP,A,E,W,XI,XN,TA,PP,QQ,WH)

ARGUMENT: GM I GRAVITATIONAL CONSTANT OF THE CENTRAL BODY

R(3) I POSITION VECTOR RELATIVE TO CENTRAL BODY

V(3) I VELOCITY VECTOR RELATIVE TO CENTRAL BODY

TFP O TIME OF FLIGHT FROM PERIAPSIS ON THE CONIC

A O SEMI-MAJOR AXIS OF THE CONIC

E O ECCENTRICITY OF THE CONIC

W O ARGUMENT OF PERIAPSIS OF THE CONIC

XI O INCLINATION OF THE CONIC TO THE REFERENCE FRAME

XN O LONGITUDE OF THE ASCENDING NODE OF THE CONIC

TA O INSTANTANEOUS TRUE ANOMALY OF THE CONIC

PP(3) O UNIT VECTOR TOWARD PERIAPSIS ON CONIC

QQ(3) O UNIT VECTOR NORMAL TO PP IN ORBITAL PLANE

WH(3) O UNIT VECTOR NORMAL TO ORBITAL PLANE

SUBROUTINES SUPPORTED: TAROPT LUNCON MULTAR EXCUTE COPINS
 NONINS CPROP VMP GUISIM NONLIN
 PULSEX GUIDM

SUBROUTINES REQUIRED: NONE

LOCAL SYMBOLS: AUXF ECCENTRIC ANOMALY (HYPERBOLIC CASE)

AVA MEAN ANOMALY (ELLIPTIC CASE)

COSEA COSINE OF THE ECCENTRIC ANOMALY (ELLIPTIC CASE)

CTA COSINE OF THE TRUE ANOMALY

C MAGNITUDE OF THE ANGULAR MOMENTUM

DIV INTERMEDIATE VARIABLE IN CALCULATION OF ECCENTRIC ANOMALY

EA	ECCENTRIC ANOMALY (ELLIPTIC CASE)
P	SEMI-LATUS RECTUM OF THE CONIC
RAD	DEGREES TO RADIANS CONVERSION CONSTANT
RD	TIME DERIVATIVE OF RADIUS
RM	MAGNITUDE OF CARTESIAN POSITION VECTOR
SINEA	SINE OF THE ECCENTRIC ANOMALY (ELLIPTIC CASE)
SINHF	HYPERBOLIC SINE OF AUXF
STA	SINE OF THE TRUE ANOMALY
TANG	INTERMEDIATE VARIABLE USED TO CALCULATE SINHF
VM	MAGNITUDE OF THE CARTESIAN VELOCITY VECTOR
Z	INTERMEDIATE VECTOR USED TO CALCULATE PP, QQ VECTORS

CAREL Analysis

CAREL converts the cartesian state (position and velocity) of a massless point referenced to a gravitational body to the equivalent conic elements about that body.

Let the cartesian state be denoted \vec{r} , \vec{v} and let the gravitational constant of the central body be μ .

The angular momentum constant c is

$$c = | \vec{r} \times \vec{v} | \quad (1)$$

The unit normal \hat{W} to the orbital plane is

$$\hat{W} = \frac{\vec{r} \times \vec{v}}{c} \quad (2)$$

The semilatus rectum p is

$$p = \frac{c^2}{\mu} \quad (3)$$

The semi-major axis a is

$$a = \frac{r}{2 - \frac{rv^2}{\mu}} \quad (4)$$

Thus $a > 0$ for elliptical motion, $a < 0$ for hyperbolic motion. The eccentricity e is

$$e = \sqrt{1 - \frac{p}{a}} \quad (5)$$

Thus $e < 1$ for elliptical motion, $e > 1$ for hyperbolic motion. The inclination of the orbit i is computed from

$$\cos i = \hat{W}_z \quad (6)$$

The longitude of the ascending node Ω is defined by

$$\tan \Omega = \frac{\hat{W}_x}{-\hat{W}_y} \quad (7)$$

The true anomaly f at the given state is computed from

$$\cos f = \frac{p - r}{e r} \quad \sin f = \frac{c \dot{r}}{\mu e} \quad (8)$$

Now define an auxiliary vector \hat{z} by

$$\hat{z} = \frac{r}{c} \vec{v} - \frac{\dot{r}}{c} \vec{r} \quad (9)$$

Then \hat{P} , the unit vector to periapsis, and \hat{Q} , the in-plane normal to \hat{P} , are defined by

$$\hat{P} = \hat{r} \cos f - \hat{z} \sin f \quad (10)$$

$$\hat{Q} = \hat{r} \sin f + \hat{z} \cos f \quad (11)$$

where $\hat{r} = \frac{\vec{r}}{r}$. The argument of periapsis ω is then computed from

$$\tan \omega = \frac{\hat{P}_z}{\hat{Q}_z} \quad (12)$$

The conic time from periapsis t_p is computed from different formulae depending upon the sign of the semi-major axis. For $a > 0$ (elliptical motion)

$$t_p = \sqrt{\frac{a^3}{\mu}} (E - e \sin E)$$

$$\cos E = \frac{e + \cos f}{1 + e \cos f} \quad \sin E = \frac{\sqrt{1 - e^2} \sin f}{1 + e \cos f} \quad (13)$$

For $a < 0$ (hyperbolic motion) the time from periapsis is

$$t_p = \sqrt{\frac{a^3}{\mu}} (e \sinh H - H)$$

$$\tanh \frac{H}{2} = \sqrt{\frac{e - 1}{e + 1}} \tan \frac{f}{2} \quad (14)$$

Reference: Battin, R. H., Astronautical Guidance, McGraw-Hill Book Co., New York, 1964.

SUBROUTINE CASCAD

PURPOSE: TO COMPUTE THE STATE TRANSITION MATRIX DEFINING STATE PERTURBATIONS OVER AN ARBITRARY TIME INTERVAL BY CASCADING DANBY MATRIZANTS OVER SEGMENTS OF THE INTERVAL USING EITHER PATCHED CONIC OR VIRTUAL MASS TWO BODY FORMULAE.

CALLING SEQUENCE: CALL CASCAD(RI, STMAT)

ARGUMENT: RI I POSITION AND VELOCITY OF VEHICLE AT BEGINNING OF TIME INTERVAL

STMAT O STATE TRANSITION MATRIX OVER DESIRED INTERVAL

SUBROUTINES SUPPORTED: PSIM

SUBROUTINES REQUIRED: CONC2 VMP

LOCAL SYMBOLS: DELTAT TIME INTERVAL USED IN A SINGLE ITERATE

DELT TIME INTERVAL OF CURRENT PROPAGATION

O1 INITIAL TIME OF ITERATE

IFLAG FLAG TO DETERMINE WHETHER ITERATION IS COMPLETED

IOS FLAG INDICATING HELIOCENTRIC OR PLANETOCENTRIC PHASE

ISP3 FLAG USED AS TRAJECTORY INTEGRATION SPHERE OF INFLUENCE STOPPING CODE

PHI CUMULATIVE STATE TRANSITION MATRIX OVER INTERVAL (T0, TK)

PSI CUMULATIVE STATE TRANSITION MATRIX OVER INTERVAL (T0, TK+1)

PTP STATE OF TARGET RELATIVE TO INERTIAL COORDINATE AT TIME TK

RAV STATE OF SPACECRAFT RELATIVE TO DOMINANT BODY FOR MATRIZANT

RHO STATE TRANSITION MATRIX OVER INTERVAL (TK, TK+1)

RS INERTIAL SPACECRAFT STATE AT TK

RSF INERTIAL SPACECRAFT STATE AT TK+1

R1 SPACECRAFT STATE RELATIVE TO VIRTUAL MASS
 AT TK

 R2 SPACECRAFT STATE RELATIVE TO VIRTUAL MASS
 AT TK+1

 SUM INTERMEDIATE VARIABLE

 TIME CUMULATIVE TRAJECTORY TIME FROM INITIAL
 TIME TO TK+1

 XHU VIRTUAL MASS MAGNITUDE AT TK

 YHU VIRTUAL MASS MAGNITUDE AT TK+1

COMMON COMPUTED:

ICL

COMMON USED:

ACC	ALNGTH	DATEJ	DELTM	DTPLAN
DTSUM	ISTM1	NTP	PMASS	RTP
RVS	TM	TRTM1	VMU	V

CASCAD Analysis

CASCAD approximates the state transition matrix $\phi_{f,o}$ defining state perturbations over an arbitrary interval $[t_o, t_f]$ by recursively computing state transition matrices over intervals $[t_o, t_1]$, $[t_o, t_2]$, ..., $[t_o, t_f]$.

The recursive formula for the $k+1$ iteration based on the k -th iteration is given by

$$\phi_{k+1,o} = \psi_{k+1,k} \phi_{k,o} \quad (1)$$

where $\psi_{k+1,k}$ is the state transition matrix for the $k+1$ -st interval $[t_k, t_{k+1}]$.

The time interval $\Delta t_{k+1} = t_{k+1} - t_k$ is determined by the position vector \vec{r}_k of the spacecraft relative to the target planet along the nominal n -body trajectory at the time t_k . Then if R_{SOI} denotes the radius of the sphere of influence of the target planet the time interval is defined by

$$\begin{aligned} \Delta t_{k+1} &= \Delta t_{\text{planet}} && \text{if } r_k \leq R_{SOI} \\ &= \Delta t_{\text{sun}} && \text{if } r_k > R_{SOI} \text{ and the } n\text{-body nominal} \\ &&& \text{trajectory propagated over } \Delta t_{\text{sun}} \text{ does} \\ &&& \text{not intersect the SOI.} \\ &= \Delta t_{SOI} && \text{if } r_k > R_{SOI} \text{ and the } n\text{-body nominal} \\ &&& \text{trajectory intersects the SOI after the} \\ &&& \text{time interval } \Delta t_{SOI} \text{ where } \Delta t_{SOI} < \Delta t_{\text{sun}}. \end{aligned}$$

where Δt_{planet} and Δt_{sun} are input parameters. For the last interval a partial step may be required so that $\Delta t_n = t_f - t_{n-1}$.

The $\psi_{k+1,k}$ matrix may be computed by either of two models. In the patch conic model the position and velocity vectors \vec{r}_k, \vec{v}_k of the spacecraft relative to the dominant body (the sun if $\Delta t_{k+1} = \Delta t_{\text{sun}}$ or Δt_{SOI} , the target planet if $\Delta t_{k+1} = \Delta t_{\text{planet}}$) at the time t_k is used to define a

conic with respect to the dominant body and the Danby matrizant over the given interval defines $\psi_{k+1,k}$ (CØNC2) .

In the virtual mass model the position and velocity vectors \vec{R}_k, \vec{V}_k are computed relative to the virtual mass and the gravitational constant used is that of the virtual mass magnitude at the time t_k . The Danby matrizant corresponding to this conic then is used to compute $\psi_{k+1,k}$ (CØNC2).

The recursive process continues until the state transition matrix over the entire interval $[t_o, t_f]$ is determined.

Reference: Danby, J.M.A., "The Matrizant of Keplerian Motion," AIAA Journal, vol 2, no 1, January, 1964.

SUBROUTINE CENTER

PURPOSE: TO CONVERT THE POSITION AND VELOCITY VECTORS OF THE GRAVITATING BODIES FROM REFERENCE BODY ECLIPTIC TO BARYCENTRIC ECLIPTIC AND STORE THEM IN THE F ARRAY.

CALLING SEQUENCE: CALL CENTER

SUBROUTINES SUPPORTED: EPHEM

SUBROUTINES REQUIRED: NONE

LOCAL SYMBOLS	BARYC	POSITION AND VELOCITY OF CENTER OF MASS RELATIVE TO EARTH. (AU, AU/DAY)
	F	ARRAY OF PLANET EPHEMERIS DATA IN AU, AU/DAY UNITS. DATA INDICATED BY THE FOLLOWING INDICES
	4*I-2,J	VELOCITY OF I-TH PLANET RELATIVE TO THE SUN (INPUT) AND RELATIVE TO THE BARYCENTER (OUTPUT)
	4*I-3,J	POSITION OF I-TH PLANET RELATIVE TO THE SUN (INPUT) AND RELATIVE TO THE BARYCENTER (OUTPUT)
	4*IM-2,J	VELOCITY OF MOON RELATIVE TO EARTH (INPUT) AND RELATIVE TO BARYCENTER OUTPUT
	4*IM-3,J	POSITION OF MOON RELATIVE TO EARTH (INPUT) AND RELATIVE TO BARYCENTER(OUTPUT)
	GEOP	POSITION AND VELOCITY OF BODIES RELATIVE TO THE EARTH. (AU, AU/DAY)
	IND	INDEX USED TO EXTRACT EARTH EPHEMERIS DATA RELATIVE TO SUN FROM F-ARRAY
	IX	INDEX OF IJ-TH GRAVITATIONAL BODY
	SUM	SUM OF GRAVITATIONAL CONSTANTS (AU**3/DAY**2)
	SUN	POSITION AND VELOCITY OF SUN RELATIVE TO EARTH (AU, AU/DAY)
COMMON COMPUTED/USED:	F	INITIAL V
COMMON USED:	NBODYI	NO
	PMASS	ZERO

CENTER Analysis

Let the state vector of position and velocity of the gravitating bodies (excluding the moon) in heliocentric ecliptic coordinates be denoted ρ_i, ω_i at some reference time. Let the index of the earth be i_E . Then the coordinates of all bodies (excluding the moon) relative to the earth is

$$\begin{aligned}\vec{r}_i &= \vec{\rho}_i - \vec{\rho}_{i_E} & i=1,n, \quad i \neq i_M \\ \vec{v}_i &= \vec{\omega}_i - \vec{\omega}_{i_E} & i=1,n, \quad i \neq i_M\end{aligned}\quad (1)$$

Let the position and velocity of the moon relative to the earth be denoted $\vec{r}_{i_M}, \vec{v}_{i_M}$.

Define the radius vector to the center of mass (in earth ecliptic coordinates) by

$$\vec{R}_{CM} = \frac{1}{M} \sum_{i=1}^n \mu_i \vec{r}_i \quad M = \sum_{i=1}^n \mu_i \quad (2)$$

Its velocity relative to the earth may then be found by differentiation.

$$\vec{v}_{CM} = \frac{1}{M} \sum_{i=1}^n \mu_i \vec{v}_i \quad (3)$$

The coordinates of all gravitating bodies relative to the center of mass may then be computed

$$\begin{aligned}\vec{R}_i &= \vec{r}_i - \vec{R}_{CM} \\ \vec{v}_i &= \vec{v}_i - \vec{v}_{CM}\end{aligned}\quad (4)$$

SUBROUTINE CONCAR

PURPOSE: TO CONVERT A CONIC STATE IN TERMS OF R, THETA, E, P, PA, QA AND GMU INTO A CARTESIAN STATE

ARGUMENT: CSTA I COSINE OF TRUE ANAMOLY AT CURRENT STATE
 E I ECCENTRICITY OF CONIC
 GMU I GRAVITATIONAL CONSTANT OF PLANET IN
 KM**3/SEC**2
 PA I UNIT VECTOR IN DIRECTION OF PERIAPSIS
 P I SEMI-LATUS RECTUM OF CONIC IN KM
 QA I UNIT VECTOR IN ORBIT PLANE IN DIRECTION
 90 DEG ADVANCED FROM PA
 RV O POSITION VECTOR OF CURRENT STATE IN KM
 R I MAGNITUDE OF POSITION VECTOR AT CURRENT
 STATE IN KM
 SNTA I SINE OF TRUE ANAMOLY AT CURRENT STATE
 VV O VELOCITY VECTOR OF CURRENT STATE IN KM/SEC
 SUBROUTINES SUPPORTED: TPPROP TPRTRG

LOCAL SYMBOLS: HOP ANGULAR MOMENTUM DIVIDED BY SEMI-LATUS
 RECTUM IN KM/SEC
 POSPA PROJECTION OF POSITION VECTOR ON PA IN KM
 POSQA PROJECTION OF POSITION VECTOR ON QA IN KM
 VELPA PROJECTION OF VELOCITY ON PA IN KM/SEC
 VELQA PROJECTION OF VELOCITY ON QA IN KM/SEC

SUBROUTINE CONC2

PURPOSE: COMPUTE STATE TRANSITION MATRIX USING ANALYTICAL
PATCHED CONIC OR ANALYTICAL VIRTUAL MASS TECHNIQUES

CALLING SEQUENCE: CALL CONC2(R,V,DELT,GMX,PSIEC)

ARGUMENT: DELT I TIME INCREMENT OVER WHICH THE STATE
TRANSITION MATRIX IS BEING COMPUTED

GMX I GRAVITATIONAL CONSTANT OF GOVERNING BODY

PSIEC O STATE TRANSITION MATRIX

R I POSITION OF THE VEHICLE RELATIVE TO THE
GOVERNING BODY

V I VELOCITY OF THE VEHICLE RELATIVE TO THE
GOVERNING BODY

SUBROUTINES SUPPORTED: PSIM CASCAD PCTM

LOCAL SYMBOLS: A SEMI-MAJOR AXIS

A1 INTERMEDIATE VARIABLE

A2 INTERMEDIATE VARIABLE

A3 INTERMEDIATE VARIABLE

AM2 INTERMEDIATE VARIABLE

C1 MAGNITUDE OF RXV

CSE COSINE OF ECCENTRIC ANOMALY

CTA COSINE OF TRUE ANOMALY

CTA2 COSINE OF TRUE ANOMALY ON ELLIPSE

DDX0 INTERMEDIATE VARIABLE

DDY0 INTERMEDIATE VARIABLE

DX0 INTERMEDIATE VARIABLE

DY0 INTERMEDIATE VARIABLE

E ECCENTRICITY

EA ECCENTRIC ANOMALY

FMI1	INTERMEDIATE VECTOR
FMI	INTERMEDIATE VECTOR
F	INTERMEDIATE VARIABLE
N	INTERMEDIATE VARIABLE
OPEC	INTERMEDIATE VECTOR
ORB	INTERMEDIATE VARIABLE
P	SEMI-LATUS RECTUM
PI	MATHEMATICAL CONSTANT
PSIOP	INTERMEDIATE STATE TRANSITION MATRIX
PV	INTERMEDIATE VECTOR
Q	INTERMEDIATE VECTOR
RQ	$R \cdot V$ DIVIDED BY MAGNITUDE OF R
RM	MAGNITUDE OF R
RRQ	$R \cdot V$
RTHD	INTERMEDIATE VARIABLE
R2	INTERMEDIATE VARIABLE
R3	INTERMEDIATE VARIABLE
SNE	SINE OF ECCENTRIC ANOMALY
SNF	SINE OF F
STA	SINE OF TRUE ANOMALY
STA2	SINE OF TRUE ANOMALY ON ELLIPSE
TIM1	INTERMEDIATE TIME
TIM2	INTERMEDIATE TIME
VM	MAGNITUDE OF V
WV	$R \times V$
X0	INTERMEDIATE VARIABLE
Y0	INTERMEDIATE VARIABLE

Z INTERMEDIATE VECTOR

COMMON USED:

EM8
TWOHALF
ZERO

ONE

THREE

TWOPI

CONC2 Analysis

CONC2 is responsible for the computation of a state transition matrix about a conic trajectory using the Danby matrizant analytic formula.

Danby has shown (see Reference 2) that the state transition matrix (or matrizant) has a particularly simple form if written in the orbital plane coordinate system. The state transition matrix Φ defined by

$$\delta x_f = \Phi(t_f, t_o) \delta x_o \quad (1)$$

where δx_f , δx_o refer to perturbations about a conic trajectory at time t_f , t_o respectively may be written in the orbital plane system

$$\Phi(t_f, t_o) = M(t_f) M^{-1}(t_o) \quad (2)$$

where $M(t)$, $M^{-1}(t)$ may be computed from the following formulae

$$M = \begin{bmatrix} \dot{X} & Y\dot{X}-h & 0 & 2X-3\tau\dot{X} & Y\dot{Y} & 0 \\ \dot{Y} & -X\dot{X} & 0 & 2Y-3\tau\dot{Y} & -Y\dot{X}-2h & 0 \\ 0 & 0 & Y & 0 & 0 & -X \\ \ddot{X} & \dot{Y}\dot{X}+Y\ddot{X} & 0 & -\dot{X}-3\tau\ddot{X} & \dot{Y}^2 + Y\ddot{Y} & 0 \\ \ddot{Y} & \dot{X}\dot{Y}-X\ddot{Y} & 0 & -\dot{Y}-3\tau\ddot{Y} & -\dot{X}\dot{Y}-Y\ddot{X} & 0 \\ 0 & 0 & \dot{Y} & 0 & 0 & -\dot{X} \end{bmatrix} \quad (3)$$

$$M^{-1} = A J M^T J^T \quad (4)$$

where $X, Y, \dot{X}, \dot{Y}, \ddot{X}, \ddot{Y}$ are evaluated at the time t
 h is the angular momentum constant
 τ is the time interval from t to some epoch (periapsis)

$$\text{and } A = \text{diag} (a/\mu, a/\mu h, 1/h, a/\mu, a/\mu h, 1/h) \quad (5)$$

$$J = \begin{bmatrix} 0 & -I \\ I & 0 \end{bmatrix} \quad (6)$$

Thus to use the Danby formulation one must determine the transformation from the reference frame to the orbital plane coordinates, compute the values of the quantities $X, Y, \dot{X}, \dot{Y}, \ddot{X}, \ddot{Y}$ and h and τ at the times t_o , t_f and then use the above equations.

Let the initial state of the conic be denoted \vec{r} , \vec{v} , the gravitational force $\vec{\mu}$, and the time interval Δt . Then the unit vectors \hat{P} in the direction of periapsis, \hat{W} in the direction of the angular momentum vector, and $\hat{Q} = \hat{W} \times \hat{P}$ defining the orbital plane coordinate system may be computed by the following conic equations

$$h = | \vec{r} \times \vec{v} | \quad (7)$$

$$\hat{W} = \frac{\vec{r} \times \vec{v}}{h} \quad (8)$$

$$\dot{r} = \frac{\vec{r} \cdot \vec{v}}{r} \quad (9)$$

$$p = \frac{h^2}{\mu} \quad (10)$$

$$a = \frac{r}{2 - rv^2/\mu} \quad (11)$$

$$e = \sqrt{1 - P/a} \quad (12)$$

$$\cos f = \frac{p - r}{er} \quad \sin f = \frac{\dot{r} h}{\mu e} \quad (13)$$

$$\vec{z} = \frac{r}{h} \vec{v} - \frac{\dot{r}}{h} \vec{r} \quad (14)$$

$$\hat{P} = \cos f \frac{\vec{r}}{r} - \sin f \vec{z} \quad (15)$$

$$\hat{Q} = \sin f \frac{\vec{r}}{r} + \cos f \vec{z} \quad (16)$$

$$\dot{f} = \frac{c}{r^2} \quad (17)$$

The transformation matrix from the original \vec{r}, \vec{v} system to the orbital plane system may then be written

$$T = \begin{bmatrix} \hat{P} & \hat{Q} & \hat{W} \end{bmatrix} \quad (18)$$

Let the true anomaly at the pertinent time (t_0 or t_f) be denoted f . Then the quantities required in (3) are written

$$\begin{aligned} X &= r \cos f & Y &= r \sin f \\ \dot{X} &= \dot{r} \cos f - r \dot{f} \sin f & \dot{Y} &= \dot{r} \sin f + r \dot{f} \cos f \\ \ddot{X} &= -\frac{\mu X}{r^3} & \ddot{Y} &= -\frac{\mu Y}{r^3} \end{aligned} \quad (19)$$

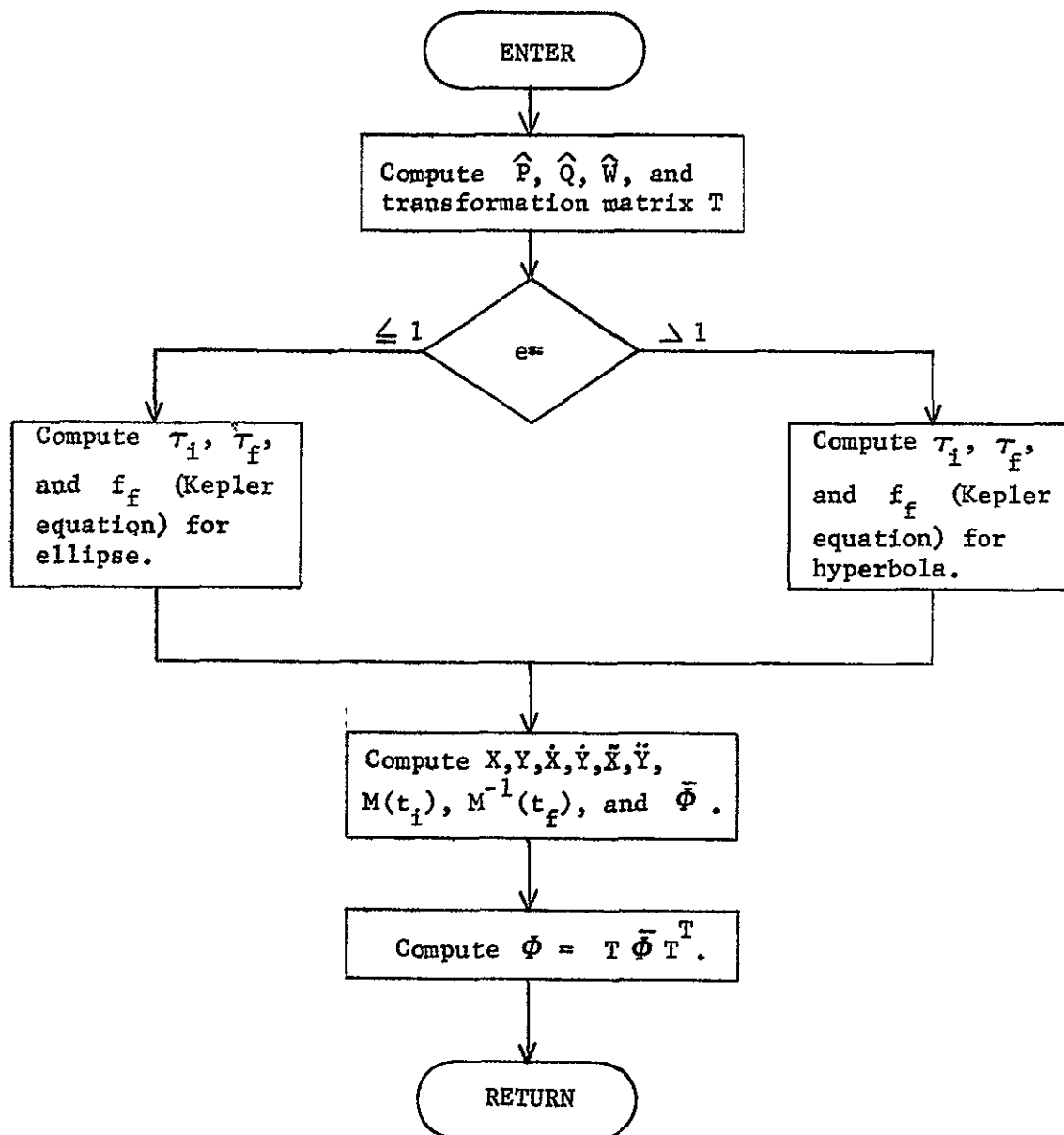
Having computed the state transition matrix $\bar{\Phi}$, corresponding to the orbital plane system by equations (2), (3), (4), it is an easy task to convert it to the normal reference system

$$\Phi = T \bar{\Phi} T^T \quad (20)$$

References: Battin, R. H., Astronautical Guidance, McGraw-Hill Book Co., New York, 1964.

Danby, J.M.A., Matrizant of Keplerian Motion, AIAA J., vol. 3, no. 4, April, 1965.

CONC2 Flow Chart



SUBROUTINE CONVRT

PURPOSE: TO COMPUTE THE GEOCENTRIC EQUATORIAL COORDINATES OF THE VEHICLE.

CALLING SEQUENCE: CALL CONVRT(R,PHI,THETA,VEL,GAMMA,SIGMA,X,Y,Z,
VX,VY,VZ)

ARGUMENT:	GAMMA	I	PATH ANGLE
	R	I	GEOCENTRIC RADIUS
	PHI	I	DECLINATION
	THETA	I	RIGHT ASCENSION
	VEL	I	VELOCITY
	SIGMA	I	AZIMUTH
	X	O	X COMPONENT OF POSITION IN GEOCENTRIC EQUATORIAL COORDINATES
	Y	O	Y COMPONENT OF POSITION IN GEOCENTRIC EQUATORIAL COORDINATES
	Z	O	Z COMPONENT OF POSITION IN GEOCENTRIC EQUATORIAL COORDINATES
	VX	O	X COMPONENT OF VELOCITY IN GEOCENTRIC EQUATORIAL COORDINATES
	VY	O	Y COMPONENT OF VELOCITY IN GEOCENTRIC EQUATORIAL COORDINATES
	VZ	O	Z COMPONENT OF VELOCITY IN GEOCENTRIC EQUATORIAL COORDINATES

SUBROUTINES SUPPORTED: DATA DATAS

LOCAL SYMBOLS:	B1	INTERMEDIATE VARIABLE
	B2	INTERMEDIATE VARIABLE
	B3	INTERMEDIATE VARIABLE
	CG	COSINE OF PATH ANGLE
	CP	COSINE OF DECLINATION
	CT	COSINE OF RIGHT ASCENSION
	SG	SINE OF PATH ANGLE

SP	SINE OF DECLINATION
ST	SINE OF RIGHT ASCENSION

CONVRT Analysis

Geocentric equatorial position and velocity components are related to geocentric radius, declination, right ascension, velocity magnitude, flight path angle, and azimuth through the following equations:

$$x = r \cos \phi \cos \theta$$

$$y = r \cos \phi \sin \theta$$

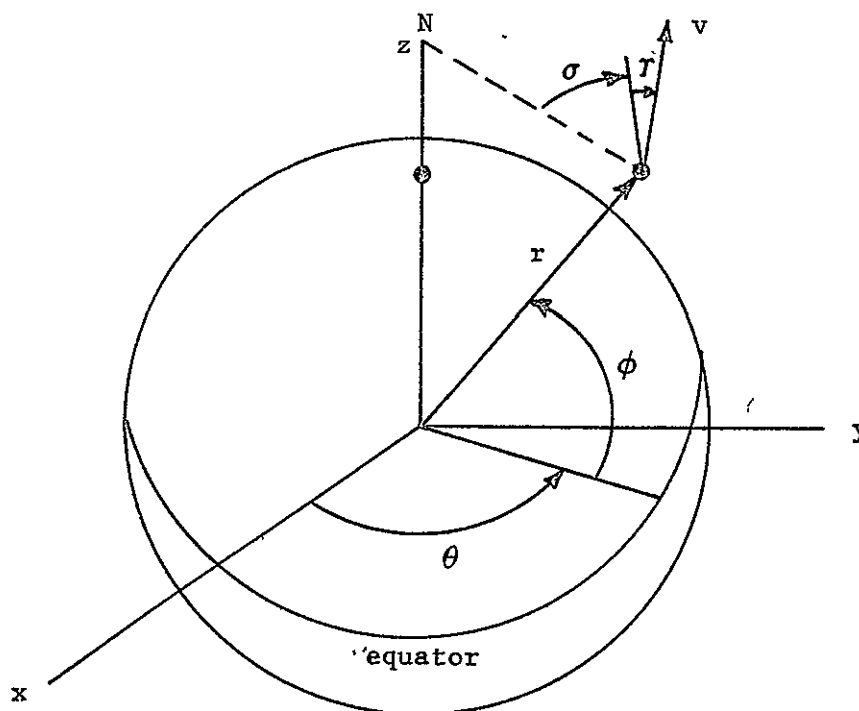
$$z = r \sin \phi$$

$$\dot{x} = v (\sin T \cos \phi \cos \theta - \cos T \sin \sigma \sin \theta - \cos T \cos \sigma \sin \phi \cos \theta)$$

$$\dot{y} = v (\sin T \cos \phi \sin \theta + \cos T \sin \sigma \cos \theta - \cos T \cos \sigma \sin \phi \sin \theta)$$

$$\dot{z} = v (\sin T \sin \phi + \cos T \cos \sigma \cos \phi)$$

The definitions of pertinent quantities are apparent in the following figure.



SUBROUTINE COPINS

PURPOSE: TO DETERMINE THE IMPULSIVE CORRECTION AND TIME REQUIRED TO INSERT FROM AN APPROACH HYPERBOLA INTO A COPLANAR ELLIPTICAL ORBIT.

CALLING SEQUENCE: CALL COPINS(GM,R,V,DA,DE,DELW,TEX,DEL V,IEX)

ARGUMENTS: GM I GRAVITATIONAL CONSTANT
 R(3) I POSITION VECTOR AT DECISION
 V(3) I VELOCITY VECTOR AT DECISION
 DE I DESIRED SEMIMAJOR AXIS
 DE I DESIRED ECCENTRICITY
 DELW I DESIRED PERIAPSIS SHIFT
 TEX O TIME FROM DECISION TO EXECUTION (SECONDS)
 DELV(3) O INSERTION VELOCITY CORRECTION
 IEX O EXECUTION CODE
 =0 EVENT IS EXECUTABLE
 =1 NO EXECUTABLE SOLUTION FOUND

SUBROUTINES SUPPORTED: INSERS

SUBROUTINES REQUIRED: CAREL ELCAR

LOCAL SYMBOLS: AA COEFFICIENT DEFINING TANGENTIAL SOLUTION FOR A
 AH HYPERBOLIC SEMIMAJOR AXIS
 ARC THE CONSTANT 180
 A1 CANDIDATE SOLUTION FOR SEMIMAJOR AXIS
 A2 CANDIDATE SOLUTION FOR SEMIMAJOR AXIS
 A TARGET SEMIMAJOR AXIS
 BB COEFFICIENT DEFINING TANGENTIAL SOLUTION FOR A
 B TANGENTIAL SOLUTION CONSTANT
 CC COEFFICIENT DEFINING TANGENTIAL SOLUTION FOR A

COE	1/E
COSTH	COS(THETA)
COSW	COS(W)
C	TANGENTIAL SOLUTION CONSTANT
DELVM	MAGNITUDE OF FINAL CORRECTION
DISC	DISCRIMINANT OF SOLUTION FOR THETA
DISK	DISCRIMINANT OF TANGENTIAL SOLUTION FOR A
DRA	DESIRED APOAPSIS RADIUS
DRP	DESIRED PERIAPSIS RADIUS
DVM	MAGNITUDE OF VELOCITY CORRECTION FOR CANDIDATE SOLUTION
DV	VELOCITY CORRECTION OF CANDIDATE SOLUTION
D	TANGENTIAL SOLUTION CONSTANT
EH	HYPERBOLIC ECCENTRICITY
ERRMAX	SCALAR ERROR ASSOCIATED WITH IMPOSSIBLE SOLUTION
ERR	SCALAR ERRORS OF CANDIDATE SOLUTIONS
ER	RADIUS ON ELLIPSE AT INSERTION
ETA	TRUE ANOMALY ON ELLIPSE AT INSERTION
E	ECCENTRICITY OF ELLIPSE
HI	INCLINATION OF HYPERBOLA
HN	ASCENDING NODE OF HYPERBOLA
HRP	HYPERBOLIC PERIAPSIS RADIUS
HR	RADIUS OF HYPERBOLA AT INSERTION
IOPT	TYPE OF SOLUTION =0 ORBITS INTERSECT =1 MUST MODIFY ORBIT TO OBTAIN SOLUTION
ISOL	INDEX OF SOLUTION

MIN	INDEX OF MINIMUM LOSS FUNCTION SOLUTION
NSOLS	NUMBER OF SOLUTIONS
PH	HYPERBOLIC SEMILATUS RECTUM
PI	THE MATHEMATICAL CONSTANT PI
PP	UNIT VECTOR TOWARD PERIAPSIS
P	ELLIPTICAL SEMILATUS RECTUM
QQ	UNIT VECTOR IN ORBIT PLANE NORMAL TO PP
RAD	DEGREE TO RADIAN TRANSFORMATION
RA	APCOPSIS RADIUS
RD	RADIUS TO DECISION STATE
REMG	MAGNITUDE OF RADIUS ON ELLIPSE AFTER INSERTION
RE	POSITION VECTOR ON ELLIPSE AFTER INSERTION
RH	POSITION ON HYPERBOLA BEFORE INSERTION
RMAG	MAGNITUDE OF RADIUS ON HYPERBOLA-BEFORE INSERTION
RP	PERIAPSIS RADIUS
SGN	PARAMETER IN TANGENTIAL SOLUTION
SINW	SIN(W)
STA	TRUE ANOMALY ON HYPERBOLA AT DECISION
SYGN	POSITIVE OR NEGATIVE SIGN IN QUADRATIC
S	INTERMEDIATE VARIABLE
TFPE	TIME FROM PERIAPSIS ON ELLIPSE
TFPH	HYPERBOLIC TIME FROM PERIAPSIS AT INSERT
THA	TRUE ANOMALY OF INSERTION ON HYPERBOLA
TINDX	TIME FROM DECISION TO EXECUTION
TIND	TIME FROM PERIAPSIS AT DECISION

VD	SPEED AT DECISION
VEMG	SPEED ON ELLIPSE AFTER INSERTION
VE	VELOCITY VECTOR ON ELLIPSE AFTER INSERTION
VH	VELOCITY VECTOR ON HYPERBOLA BEFORE INSERTION
VHAG	SPEED ON HYPERBOLA BEFORE INSERTION
WH	HYPERBOLIC ARGUMENT OF PERIAPSIS
WW	UNIT NORMAL TO ORBITAL PLANE
WX	ARGUMENT OF PERIAPSIS OF ELLIPSE
W	PERIAPSIS SHIFT
X	CONSTANT IN EQUATION DEFINING HYPERBOLIC TRUE ANOMALY AT INSERTION
Y	CONSTANT IN EQUATION DEFINING HYPERBOLIC TRUE ANOMALY AT INSERTION
Z	CONSTANT IN EQUATION DEFINING HYPERBOLIC TRUE ANOMALY AT INSERTION

COPINS Analysis:

COPINS determines the impulsive correction and time required to insert from an approach hyperbola into a coplanar elliptical orbit. The approach hyperbola is specified by a planetocentric state \vec{r}, \vec{v} at a decision time t_d . The desired elliptical orbit is prescribed by input parameters $a, e, \Delta\omega$ where a and e are the semi-major axis and eccentricity of the desired ellipse and $\Delta\omega$ is the angle (measured counter clockwise) from the hyperbolic periapsis to the periapsis of the desired orbit. The situation is illustrated in Figure 1.

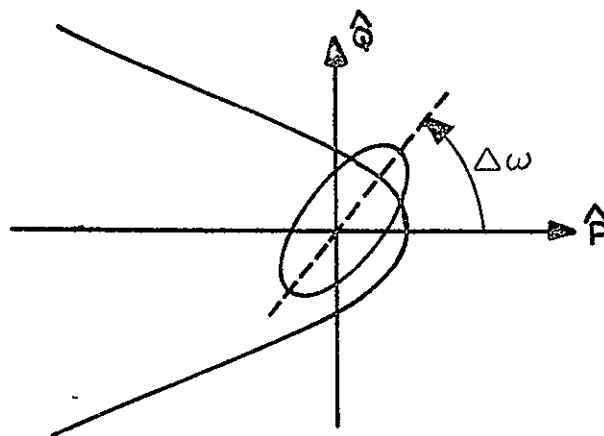


Figure 1. Approach Hyperbola and Desired Orbit

The planetocentric ecliptic state \vec{r}, \vec{v} at the time of decision t_d is first converted to Keplerian elements. $(a_H, e_H, i_H, \Omega_H, t_{Hd})$ via subroutine CAREL where t_{Hd} is the time from periapsis (negative on the approach ray). The angle f_∞ between the hyperbolic periapsis and the approach asymptote \hat{S} is computed from

$$\cos f_\infty = \frac{1}{e}, \quad 0 < f_\infty < 90^\circ \quad (1)$$

Thus the angle ω between the hyperbolic periapsis and the desired elliptical periapsis is given by

$$\omega = \Delta\omega \quad (2)$$

The hyperbola and ellipse may therefore be described in the PQ plane by standard conic formula, specifically,

$$r_H = \frac{p_H}{1 + e_H \cos \theta} \quad (3)$$

$$r_E = \frac{p_E}{1 + e_E \cos(\theta - \omega)}$$

where θ is measured counter-clockwise from \hat{P} and p_H, p_E are the semi-latus rectum of the hyperbola and ellipse respectively. Obviously if an angle of intersection θ^* is known, the states on both conics (\vec{r}^*, \vec{v}_H^*) and (\vec{r}^*, \vec{v}_E^*) may be computed from conic formulae and the desired impulsive correction is given by

$$\Delta \vec{v} = \vec{v}_E^* - \vec{v}_H^* \quad (4)$$

Likewise the time from periapsis to the intersection point t^* may be computed using hyperbolic formula and therefore the time from decision to execution is given by

$$\Delta t = t^* - t_d \quad (5)$$

Thus the coplanar insertion problem reduces to the determination of the optimal angle θ^* for the impulsive maneuver.

From (3) the values of θ for which $r_H = r_E$ are given by

$$\cos \theta = \frac{-xy \pm z\sqrt{D}}{y^2 + z^2} \quad (6)$$

where

$$\begin{aligned} x &= p_H - p_E \\ y &= p_H e_E \cos \omega - p_E e_H \\ z &= p_H e_E \sin \omega \\ D &= y^2 + z^2 - x^2 \end{aligned} \quad (7)$$

If the discriminant $D \geq 0$ there are at most two real non-extraneous solutions θ_1, θ_2 such that $r_E(\theta) = r_H(\theta)$. Note that the angle θ may not lie in the region inside the approach and departure asymptotes. If there are two solutions, both Δv 's are computed by (4) and the minimum Δv transfer is selected.

If $D < 0$, the applied hyperbola and the desired orbit do not intersect and there is no impulsive transfer between the two conics. In such a case the desired elements a_E and e_E are modified to determine the "best" tangential solution possible. Three different modifications are tested:

- (1) Vary r_a while holding r_p at the desired value.
- (2) Vary r_p while holding r_a at the desired value.
- (3) Vary a_E while holding e_E at the desired value.

The three modification schemes are illustrated in Figure 2 where the original nonintersecting orbit is shown by the broken lines.

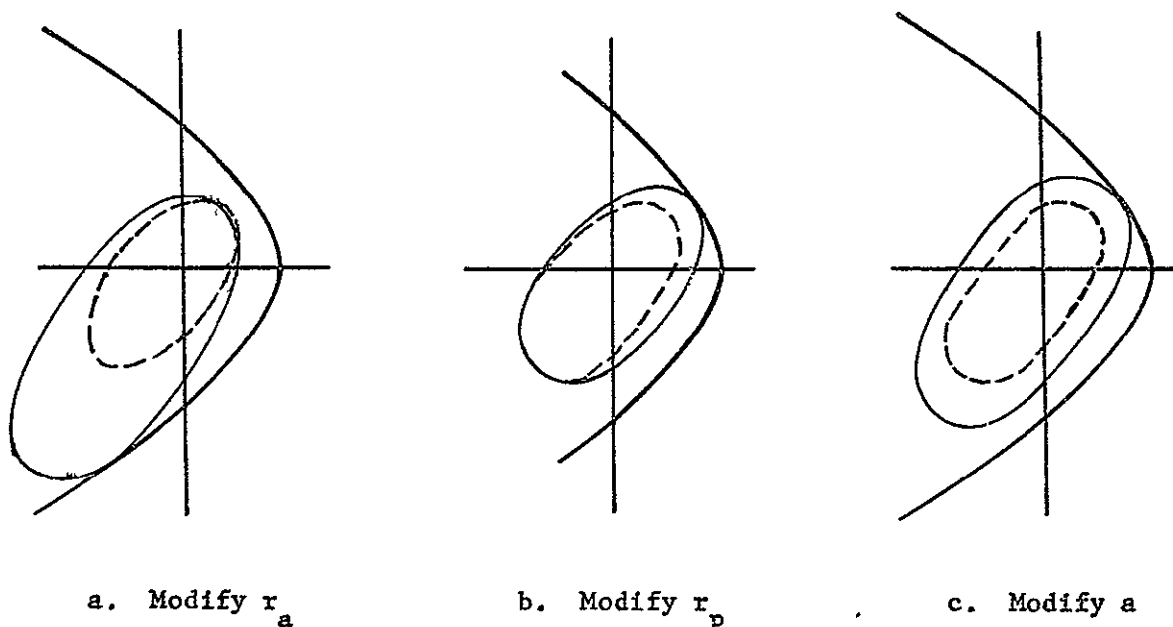


Figure 2. Candidate Orbit Modifications

It is desired to modify the "a" and the "e" of the desired orbit to achieve the tangential configurations. From (6) it is obvious that a necessary condition for a tangential solution is given by $D=0$. Using (7) D may be written

$$D = p_H^2 (e_E^2 - 1) + p_E^2 b + 2p_H p_E - c p_E e_E$$

where $b = e_H^2 - 1$

$$c = 2p_H e_H \cos \omega \quad (8)$$

where it is observed the approach hyperbola is fixed and it is desired not to vary the ω of the desired ellipse so that subsequent apsidal rotations are avoided.

Modification Option 1: Rewriting (8a) in terms of a and r_p leads to

$$\begin{aligned} a^2 D = & (4 r_p^2 b + 4 r_p p_H - 2 r_p c) a^2 \\ & + (-2 p_H^2 r_p - 4 r_p^3 b - 2 p_H r_p^2 + 3 r_p^2 b) a \\ & + (p_H^2 r_p^2 + r_p^4 b - c r_p^3) \end{aligned} \quad (9)$$

Now if D is set equal to 0, r_p held at its desired value, and the resulting quadratic solved for "a", the solution will correspond to the tangential solution which holds r_p constant. If $a \leq 0$ or imaginary, the solution is disregarded. The modified eccentricity is of course defined by

$$e = 1 - \frac{r_p}{a} \quad (10)$$

Modification Option 2: Rewriting (8a) in terms of a and r_a leads to

$$\begin{aligned} a^2 D = & (4 r_a^2 b + 4 r_a p_H + 2 r_a c) a^2 \\ & + (-2 p_H^2 r_a - 4 r_a^3 b - 2 p_H r_a^2 - 3 r_a^2 c) a \\ & + (p_H^2 r_a^2 + r_a^4 b + c r_a^3) \end{aligned} \quad (11)$$

For computational purposes the similarity between (9) and (11) may be exploited. Again setting $D = 0$ and holding r_a at its desired value, the value of "a" may be determined which specifies the tangential solution holding r_a constant. Having determined a realistic value of "a", the corresponding eccentricity is given by

$$e = \frac{r_a}{a} - 1 \quad (12)$$

Modification Option 3: Rewriting (8a) in terms of a and e_E leads to

$$\begin{aligned} D = & (d^2 b) a^2 + (2 p_H d - c d e_E) a - d p_H^2 \\ d = & (1 - e_E^2) \end{aligned} \quad (13)$$

Setting $D = 0$ and solving for "a" while holding e_E at its desired value then defines the option 3 solution.

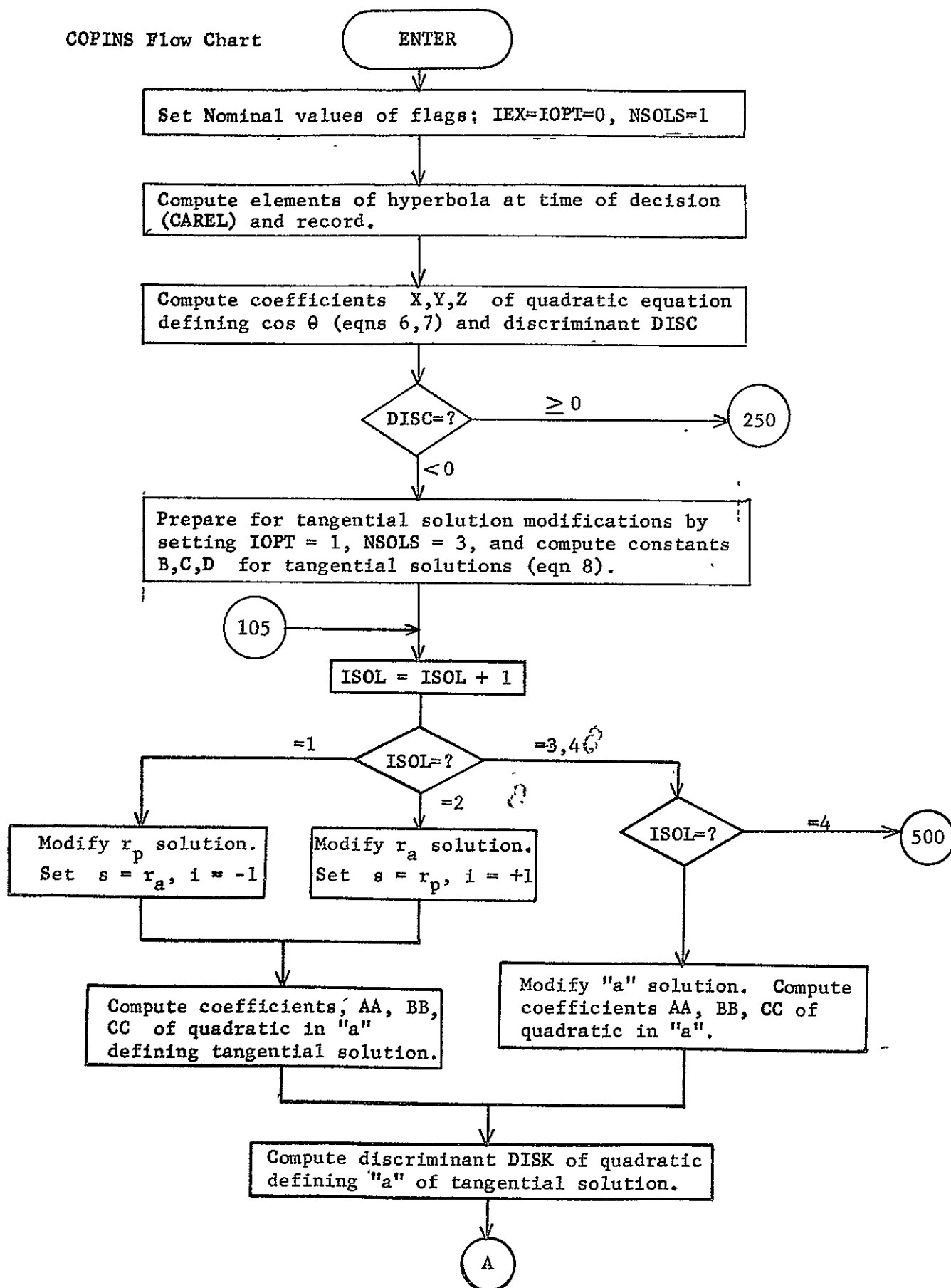
To determine the "best" modified orbit from the three candidate options a rather arbitrary scheme is used. A scalar error is assigned to each option according to a weighting factor and the difference between the desired and achieved values of the periapsis and apoapsis radii:

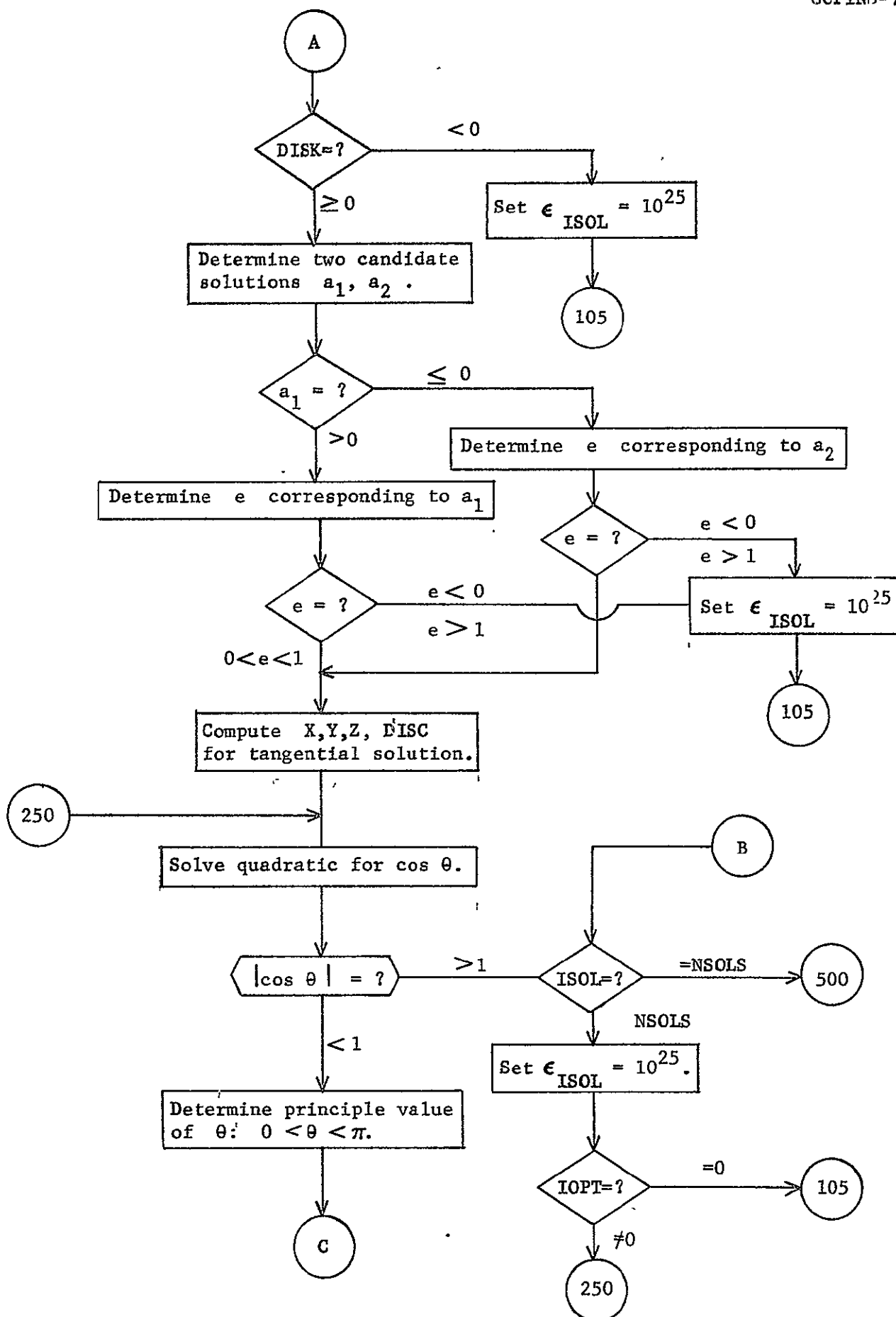
$$E_1 = W_1 (|\Delta r_a| + |\Delta r_p|)$$

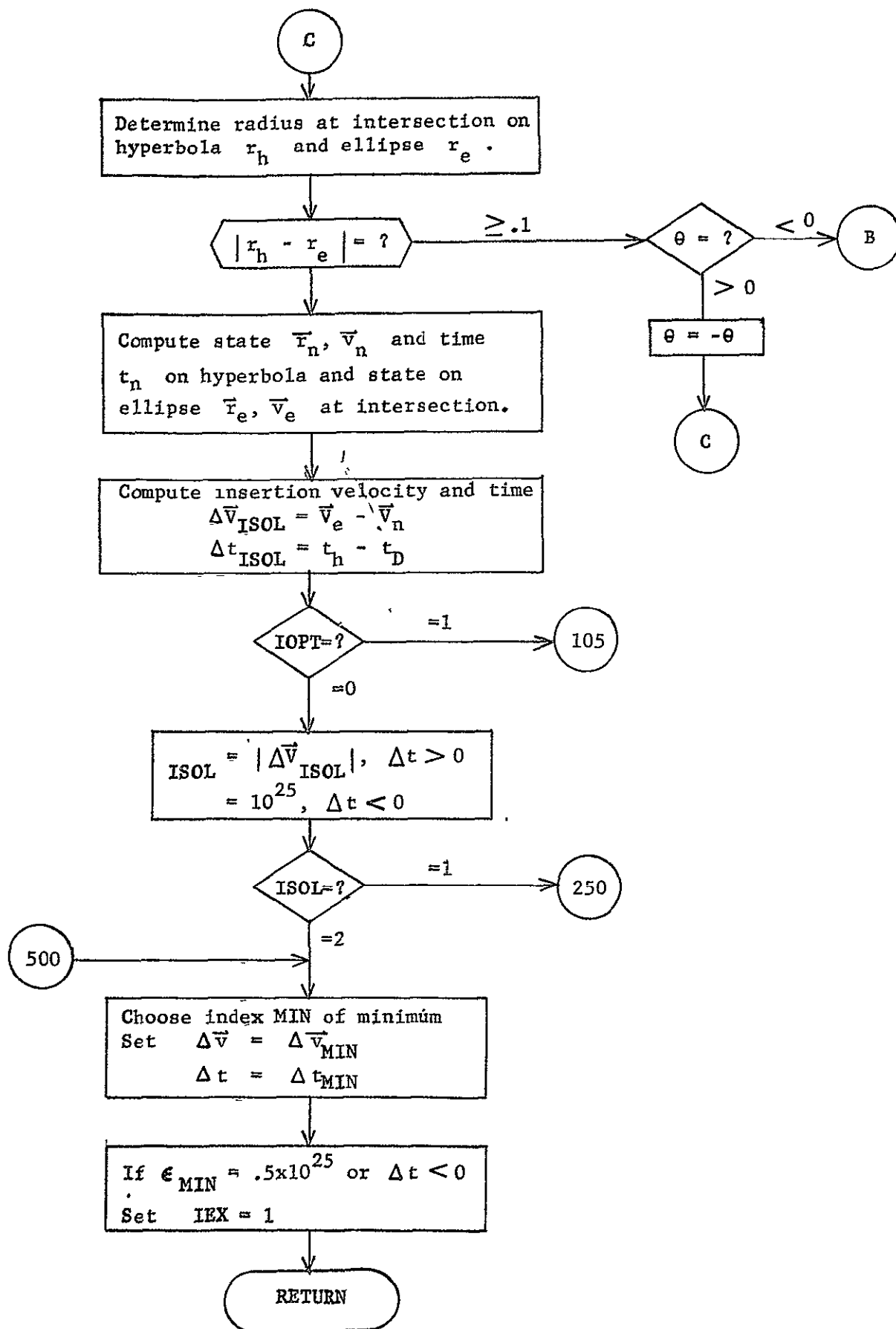
where the scalar factor W_1 is set to 1,2,3 respectively for the three options. Thus the preferred strategy is the one which requires a correction only at apoapsis while the least desired scheme requires subsequent corrections both at periapsis and apoapsis.

Having determined orbital elements that necessarily lead to a tangential solution, (6) may now be used to compute the angle of intersection θ .

COPINS Flow Chart







SUBROUTINE CORREL

PURPOSE: CONVERT COVARIANCE MATRIX PARTITIONS TO CORRELATION MATRIX PARTITIONS AND STANDARD DEVIATIONS AND WRITE THEM OUT

CALLING SEQUENCE: CALL CORREL(PP,CXXSP,PSP,CXUP,U0,CXVP,V0,CXSUP,CXSVP)

ARGUMENT:

PP	I	POSITION/VELOCITY COVARIANCE MATRIX
CXXSP	I	CORRELATION BETWEEN SOLVE-FOR PARAMETERS AND POSITION/VELOCITY STATE
PSP	I	SOLVE-FOR PARAMETER COVARIANCE MATRIX
CXUP	I	CORRELATION BETWEEN POSITION/VELOCITY STATE AND DYNAMIC CONSIDER PARAMETERS
U0	I	DYNAMIC CONSIDER PARAMETER COVARIANCE MATRIX
CXVP	I	CORRELATION BETWEEN POSITION/VELOCITY STATE AND MEASUREMENT CONSIDER PARAMETERS
V0	I	MEASUREMENT CONSIDER PARAMETER COVARIANCE MATRIX
CXSUP	I	CORRELATION BETWEEN SOLVE-FOR PARAMETERS AND DYNAMIC CONSIDER PARAMETERS
CXSVP	I	CORRELATION BETWEEN SOLVE-FOR PARAMETERS AND MEASUREMENT CONSIDER PARAMETERS

SUBROUTINES SUPPORTED: PRINT4 SETEVS GUISIM PRESIM PRNTS4
PRINT3 SETEVN GUIDM PRED PRNTS3

LOCAL SYMBOLS:

DUM	INVERSE OF SQUARE ROOT OF DIAGONAL ELEMENTS IN DYNAMIC AND MEASUREMENT CONSIDER COVARIANCE PARTITIONS
IEND	COUNTER INDICATING TOTAL NUMBER OF AUGMENTED STATE VARIABLES
ROW	INTERMEDIATE COMPUTATION AND OUTPUT VECTOR
SQP	INVERSE OF THE SQUARE ROOT OF DIAGONAL ELEMENTS IN VEHICLE AND SOLVE-FOR COVARIANCE PARTITIONS
ZZ	STANDARD DEVIATION

COMMON USED: KPRINT NDIM1 NDIM2 NDIM3 ONE

SUBROUTINE DATA

PURPOSE: TO READ INPUT DATA, TRANSLATE THIS DATA INTO PROPER INTERNAL VALUES, ASSIGN VALUES TO UNSPECIFIED NAMELIST VARIABLES, SET NECESSARY INITIAL VALUES, COMPUTE DIMENSIONS OF STATE TRANSITION, OBSERVATION, AND COVARIANCE MATRIX PARTITIONS, ORDER MEASUREMENT AND EVENT SCHEDULES, AND PRINT OUT INITIAL CONDITIONS IN THE ERRAN PROGRAM.

CALLING SEQUENCE: CALL DATA

ARGUMENT: NONE

SUBROUTINES SUPPORTED: ERRON

SUBROUTINES REQUIRED:	CONVRT	EPHEM	GHA	ORB	PECEQ
	TIME	TRANS	DATA1	GOATA	ELCAR

LOCAL SYMBOLS:	AI	INCLINATION
	AMIN	INTERMEDIATE VARIABLE
	ANODE	LONGITUDE OF ASCENDING NODE
	A	SEMIMAJOR AXIS
	DGTR	DEGREE TO RADIAN CONVERSION
	DUM1	INTERMEDIATE STORAGE ARRAY
	DUM	INTERMEDIATE STORAGE ARRAY
	D	INTERMEDIATE JULIAN DATE
	DATE	ARRAY CONTAINING FINAL JULIAN DATE
	EARTH	CALENDAR DATE AT WHICH EARTHS ORBITAL ELEMENTS WILL BE CALCULATED
	E	ECCENTRICITY
	FNDT	DATE OF FINAL TIME
	GAMMA	PATH ANGLE
	GM	GRAVITATIONAL CONSTANT OF CENTRAL BODY
	IDAY	CALENDAR DAY OF FINAL TIME
	IHR	CALENDAR HOUR OF FINAL TIME
	IMIN	CALENDAR MINUTES OF FINAL TIME

IMO	CALENDAR MONTH OF FINAL TIME
IPMN	FLAG FOR MAIN PROBE MEASUREMENT NOISE
ISMN	FLAG FOR MINI-PROBE MEASUREMENT NOISE
IYR	CALENDAR YEAR OF FINAL TIME
JUPITER	CALENDAR DATE AT WHICH ORBITAL ELEMENTS OF JUPITER WILL BE CALCULATED
LDAY	CALENDAR DAY OF INITIAL TIME
LHR	CALENDAR HOURS OF INITIAL TIME
LMIN	CALENDAR MINUTES OF INITIAL TIME
LMO	CALENDAR MONTH OF INITIAL TIME
LYR	CALENDAR YEAR OF INITIAL TIME
MARS	CALENDAR DATE AT WHICH ORBITAL ELEMENTS OF MARS WILL BE CALCULATED
MERCURY	CALENDAR DATE AT WHICH ORBITAL ELEMENTS OF MERCURY WILL BE CALCULATED
MOON	CALENDAR DATE AT WHICH ORBITAL ELEMENTS OF EARTHS MOON WILL BE CALCULATED
NENT	NUMBER OF ENTRIES IN MEASUREMENT SCHEDULE
NEPTUNE	CALENDAR DATE AT WHICH ORBITAL ELEMENTS OF NEPTUNE WILL BE CALCULATED
OME	ARGUMENT OF PERIAPSIS
PHIT	DECLINATION
PLUTO	CALENDAR DATE AT WHICH ORBITAL ELEMENTS OF PLUTO WILL BE CALUCLATED
PRD	INITIAL SEMI-LATUS RECTUM OF SPACECRAFT ORBIT
RDS	GEOCENTRIC RADIUS OF VEHICLE
RD	MAGNITUDE OF POSITION VECTOR
SATURN	CALENDAR DATE AT WHICH ORBITAL ELEMENTS OF SATURN WILL BE CALCULATED

SEC INTERMEDIATE CALENDAR SECONDS
 SECI CALENDAR SECONDS AT FINAL TIME
 SECL CALENDAR SECONDS AT INITIAL TIME
 SIGMA AZIMUTH
 TA TRUE ANOMALY OF INSTANTANEOUS POSITION AND VELOCITY
 THETA RIGHT ASCENSION
 URANUS CALENDAR DATE AT WHICH ORBITAL ELEMENTS OF URANUS WILL BE CALCULATED
 VEL INJECTION VELOCITY RELATIVE TO EARTH
 VENUS CALENDAR DATE AT WHICH ORBITAL ELEMENTS OF VENUS WILL BE CALCULATED
 VL MAGNITUDE OF VELOCITY VECTOR
 VUNIT INTERMEDIATE VELOCITY CONVERSION FACTOR

COMMON COMPUTED/USED:	ACCND	ACC	ALNGTH	CXSU	CXSV
	CXU	CXV	CXXS	DELAXS	DELECC
	DELICL	DELMA	DELMUP	DELMUS	DELNOO
	DELTP	DELW	DTMAX	DTPLAN	OTSUN
	EM1	EM4	EM5	EM6	EM7
	EM8	EPS	EP50	FACP	FACV
	FNTM	FOP	FOV	IAUGIN	IBARY
	ICDQ3		ICCOORD	ICCOOR	IGNF
	IEIG	IEPHEM	IEVNT	IHYP1	IMNF
	INPR		IPRINT	IPRT	ISP2
	ISTMC	ISTM1	KPRINT	MNCN	NBOD
	NDACC	NDIM1	NDIM2	NDIM3	NEV10
	NEV11	NEV1	NEV2	NEV3	NEV4
	NEV5	NEV6	NEV7	NEV9	NEV
	NMN	NO	NST	NTMC	ONE
	PS	P	RAD	SAL	SIGALP
	SIGBET	SIGPRO	SIGRES	SLAT	SLON
	SSS	TEV	TMN	TM	TRTM1
	TWO	T	UST	UO	VST
	VO	WST	XI	XP	ZERO
	ACTPP	DELV	FACTR	IAUGH	IBAG
	IBIAS	IDENS	IGAIN	IGEN	IGUID
	IPCSK	INPR	IPGCK	IPQ	IPROPI
	IUTC	ISAO	LKTP	LKLP	NBODYI
	NDIM4	NENT1	NENT2	NEV4	NMNP
	PMN	PROBI	PSIGA	PSIGB	PSIGK
	PSIGS	PULMAG	PULMAS	RAD	SMN

	S0	TINJ	VTANGM	XDELV	XFAC
COMMON COMPUTED:	BDRSI1	BDRSI2	BDRSI3	BDTSI1	BDTSI2
	BDTSI3	BSI1	BSI2	BSI3	CXSUB
	CXSUG	CXSVB	CXSVG	CXUB	CXUG
	CXVB	CXVG	CXXSB	CXXSG	DELTm
	EM13	EM2	EM3	EM9	EM
	HALF	IAUGDC	IAUGMC	IAUG	ICA1
	ICA2	ICA3	ICL2	ICL	
	INCMT	INITAL		ISOI1	ISOI2
	ISOI3	ISPH	ITR	MCNTR	MCODE
	NAE	NBODYI	NEV8	NGE	NPE
	NQE	OMEGA	PB	PG	PS8
	PSG	RCA1	RCA2	RCA3	RSOI1
	RSOI2	RSOI3	TCA1	TCA2	TCA3
	TG	THREE	TIMINT	TRTMB	TSOI1
	TSOI2	TSOI3	TWOPI	VSOI1	VSOI2
	VSOI3	XB	XF	XG	XSL
	XU	XV			
COMMON USED:	AINC7	ANODE7	DATEJ	DNCN	ECC7
	ELMNT	EVNM	F	G	HP7
	IPROB	MNNAME	NB	NLP	NTP
	PERP7	PI	PLANET	PMASS	P7
	TAU7	TPT2	XLAB	XNM	

PROGRAM DATAS

PURPOSE: TO READ INPUT DATA, TRANSLATE THIS DATA INTO PROPER
INTERNAL VALUES, ASSIGN VALUES TO UNSPECIFIED NAMELIST
VARIABLES, SET NECESSARY INITIAL VALUES, COMPUTE
DIMENSIONS OF STATE TRANSITION, OBSERVATION, AND
COVARIANCE MATRIX PARTITIONS, ORDER MEASUREMENT AND EVENT
SCHEDULES, AND PRINT OUT INITIAL CONDITIONS IN THE SIMUL
PROGRAM.

CALLING SEQUENCE: CALL DATAS

SUBROUTINES SUPPORTED: MAIN

SUBROUTINES REQUIRED: CONVRT DATAIS ELCAR EPHEM ORB
 PECEQ TIME TRANS

LOCAL SYMBOLS	AI	INITIAL INCLINATION OF SPACECRAFT ORBIT
	ANODE	INITIAL LONGITUDE OF ASCENDING NODE OF SPACECRAFT ORBIT
	A	INITIAL SEMI-MAJOR AXIS OF SPACECRAFT ORBIT
	DATE	ARRAY CONTAINING FINAL JULIAN DATE
	DUM1	PLANETO-CENTRIC ECLIPTIC SPACECRAFT STATE
	DUM	COORDINATE TRANSFORMATION FROM PLANETO- CENTRIC EQUATORIAL TO PLANETO-CENTRIC ECLIPTIC COORDINATES
	D	JULIAN DATE AT LAUNCH
	EARTH	CALENDAR DATE AT WHICH ORBITAL ELEMENTS OF EARTH WILL BE CALCULATED
	E	INITIAL ECCENTRICITY OF SPACECRAFT ORBIT
	FNDT	FINAL JULIAN DATE
	GAMMA	INJECTION PATH ANGLE
	GM	GRAVITATIONAL CONSTANT OF TARGET PLANET
	IDAY	DAY OF FINAL COMPUTATION
	IHR	HOUR OF FINAL COMPUTATION
	IMIN	MINUTE OF FINAL COMPUTATION
	IMO	MONTH OF FINAL COMPUTATION

IYR	YEAR OF FINAL COMPUTATION
JUPITER	CALENDAR DATE AT WHICH ORBITAL ELEMENTS OF JUPITER WILL BE CALCULATED
LDAY	LAUNCH DAY
LHR	LAUNCH HOUR
LMIN	LAUNCH MINUTE
LMO	LAUNCH MONTH
LYR	LAUNCH YEAR
MARS	CALENDAR DATE AT WHICH ORBITAL ELEMENTS OF MARS WILL BE CALCULATED
MERCURY	CALENDAR DATE AT WHICH ORBITAL ELEMENTS OF MERCURY WILL BE CALCULATED
MOON	CALENDAR DATE AT WHICH ORBITAL ELEMENTS OF MOON WILL BE CALCULATED
MENT	NUMBR OF ENTRIES IN MEASUREMENT SCHEDULE
NEPTUNE	CALENDAR DATE AT WHICH ORBITAL ELEMETNS OF NEPTUNE WILL BE CALCULATED
OME	INITIAL ARGUMENT OF PERIAPSIS OF SPACE-CRAFT ORBIT
PHIT	INJECTION DECLINATION
PLUTO	CALENDAR DATE AT WHICH ORBITAL ELEMENTS OF PLUTO WILL BE CALCULATED
PRD	INITIAL SEMI-LATUS RECTUM OF SPACECRAFT ORBIT
RDS	EARTH-CENTERED INJECTION RADIUS
RD	DUMMY VARIABLE
SATURN	CALENDAR DATE AT WHICH ORBITAL ELEMENTS OF SATURN WILL BE CALCULATED
SECI	SECOND OF FINAL COMPUTATION
SECL	LAUNCH SECOND
SEC	SECOND OF CALENDAR DATE AT WHICH ORBITAL

ELEMENTS OF A PLANET WILL BE CALCULATED

SIGMA INJECTION AZIMUTH
 TA INITIAL SPACECRAFT TRUE ANOMALY
 THETA INJECTION RIGHT ASCENSION
 URANUS CALENDAR DATE AT WHICH ORBITAL ELEMENTS OF
 URANUS WILL BE CALCULATED
 VEL INJECTION VELOCITY RELATIVE TO EARTH
 VENUS CALENDAR DATE AT WHICH ORBITAL ELEMENTS OF
 VENUS WILL BE CALCULATED
 VL DUMMY VARIABLE
 VUNIT VELOCITY CONVERSION FACTOR

COMMON COMPUTED/USED:

AALP	ABET	ACCND	ACC1	ACC
ADEVX	ALNGTH	APRO	ARES	BIA
CXSU	CXSV	CXU	CXV	CXXS
DAB	DEB	DELAXS	DELECC	DELICL
DELMA	DELMUP	DELMUS	DELNOD	DELTP
DELM	DIB	DMAB	DMUPB	DMUSB
DNOB	DTMAX	DTPLAN	DTSUN	DWB
EM1	EM4	EM5	EM6	EM7
EM8	EPS	EP50	FACP	FACV
FNTM	FOP	FOV	IAMNF	IAUGIN
IBARY	ICDT3	ICCOORD	IC00R	IDNF
IEIG	IEPHEN	IHYR1	IMNF	INPR
IOPT7	IPRINT	IPRT	ISP2	ISTMC
ISTM1	KPRINT	MNCN	NBOD1	NBOD
NB1	NDACC	NDIM1	NDIM2	NDIM3
NEV10	NEV11	NEV1	NEV2	NEV3
NEV4	NEV5	NEV6	NEV7	NEV9
NO	NST	NTMC	ONE	PS
P	RAD	SAL	SIGALP	SIGBET
SIGPRO	SIGRES	SLAT	SLB	SLON
SSS	TM	TRTM1	TTIM1	TTIM2
TWO	T	UNMAC	UST	U0
VST	V0	HST	XI	XP
ZERO				

COMMON COMPUTED:

ADEVXS	BDRSI1	BDRSI2	BDRSI3	BDTSI1
BDTSI2	BDTSI3	BSI1	BSI2	BSI3
DELTM	EM13	EM2	EM3	EM9
HALF	IAUGDC	IAUGMC	IAUG	ICA1
ICA2	ICA3	ICDQ3	ICL2	ICL
INCHT	INITAL	ISOI1	ISOI2	ISOI3
ISPH	ITR	NBODYI	NEV8	RCA1

RCA2	RCA3	RSOI1	RSOI2	RSOI3
TCA1	TCA2	TCA3	TEV	THREE
TIMINT	TRTMB	TSOI1	TSOI2	TSOI3
TWOPI	VSOI1	VSOI2	VSOI3	XI1
XSL	XU	XV	ZI	

COMMON USED:

AINC7	ANODE7	AVARM	DATEJ	DNCN
ECC7	ELMNT	F	HP7	IPROB
NAF6	NB	NLP	NTP	PERP7
PI	PLANET	PMASS	P7	TAU7
TPT2	T1	T2	T3	T4
T5	T6	T7	XNM	

SUBROUTINE DATA1

PURPOSE: TO CONTINUE THE INITIALIZATION PROCESS DESCRIBED UNDER DATA.

CALLING SEQUENCE: CALL DATA1(NENT)

ARGUMENT: NENT I NUMBER OF CARDS IN THE MEASUREMENT SCHEDULE

SUBROUTINES REQUIRED: GHA SKEDM

SUBROUTINES SUPPORTED: DATA,

LOCAL SYMBOLS: AMIN INTERMEDIATE VARIABLE
 AP INTERMEDIATE TIME ARRAY
 DGTR DEGREE TO RADIAN CONVERSION
 ICNT COUNTER ON MEASUREMENT SCHEDULE CARDS
 IGO INTERNAL FLAG
 IROW INTERMEDIATE ROW INDEX
 MEAS MEASUREMENT CODES
 NOUT DIMENSION OF AUGMENTED COVARIANCE MATRIX

COMMON COMPUTED/USED:	IEVNT	NEV	NMN	SLAT	SLON
	TEV	TMN	T1	T2	T3
	T4	T5	T6	T7	T

COMMON COMPUTED:	CXSUB	CXSUG	CXSVB	CXSVG	CXUB
	CXUG	CXVB	CXVG	CXXSB	CXXSG
	EM	EPS	IIPOL	IPOL	MCNTR
	MCODE	NAE	NGE	NPE	NQE
	OMEGA	PB	PG	PSB	PSG
	TG	XB	XF	XG	

COMMON USED:	CXSU	CXSV	CXU	CXV	CXXS
	DATEJ	DNCN	EM7	EM8	EP50
	EVNM	FACP	FACV	FNTM	ICDQ3
	IDNF	IGUID	IHYP1	IMNF	ISTMC
	MNCN	MNNAME	NDIM1	NDIM2	NDIM3
	NEV1	NEV2	NEV3	NEV4	NEV5
	NEV6	NEV7	NST	ONE	PS
	P	RAD	SAL	TPT2	TRTM1
	U0	V0	XI	ZERO	
	MCODE1	MCODE2	NENT1	NENT2	NMNP
	TMN1	TMN2	T6	T7	

SUBROUTINE DATA1S

PURPOSE: TO CONTINUE THE INITIALIZATION PROCESS DESCRIBED UNDER DATAS.

CALLING SEQUENCE: CALL DATA1S(NENT)

ARGUMENT: NENT I NUMBER OF CARDS IN THE MEASUREMENT SCHEDULE

SUBROUTINES SUPPORTED: DATAS

SUBROUTINES REQUIRED: GHA

LOCAL SYMBOLS:

AMIN	INTERMEDIATE VARIABLE
AP	INTERMEDIATE TIME ARRAY
ICNT	COUNTER ON MEASUREMENT SCHEDULE CARDS
IROW	INTERMEDIATE ROW INDEX
MEAS	MEASUREMENT CODES
NOUT	DIMENSION OF AUGMENTED COVARIANCE MATRIX
PARAM	ARRAY OF AUGMENTED BIASES
SCHED	ARRAY OF TIMES IN MEASUREMENT SCHEDULE

COMMON COMPUTED/USED:	ADEVXS	IEVNT	NEV	NMN	SLAT
	SLB	SLON	TEV	TMN	T1
	T2	T3	T4	T5	T6
	T7	T			

COMMON COMPUTED:	ADEVSB	ADEVXB	CXSUB	CXSUG	CXSVB
	CXSVG	CXUB	CXUG	CXVB	CXVG
	CXXSB	CXXSG	EDEVXS	EDEVX	EM
	EPS	IIPOL	IPOL	MCNTR	MCODE
	NAE	NGE	NPE	NQE	OMEGA
	PB	PG	PSB	PSG	TG
	XB	XF	XG	XI1	ZI

COMMON USED:	ACC1	ADEVX	AVARM	BIA	CXSU
	CXSV	CXU	CXV	CXXS	DAB
	DATEJ	DEB	DIB	DNAB	DMUPB
	DMUSB	DNCN	DNQB	DNQ	EM7
	EM8	EP50	EVNM	FACP	FACV
	FMTM	IAHNF	IAUGIN	ICDQ3	ICDT3
	IDNF	IHYP1	IMNF	ISTMC	MNCN
	MNNAME	NBOD1	NB1	NDIM1	NDIM2
	NDIM3	NEV1	NEV2	NEV3	NEV4

NEV5	NEV6	NEV7	NST	ONE
PLANET	PS	P	RAD	SAL
TPT2	TRYM1	TTIM1	TTIM2	UNMAC
U0	V0	XDUM	XI	ZERO

SUBROUTINE DESENT

PURPOSE: TO COMPUTE A CORRECTION TO AN INITIAL VELOCITY BY THE
 STEEPEST DESCENT OR CONJUGATE GRADIENT TECHNIQUES FOR
 USE BY TARGET.

CALLING SEQUENCE: CALL DESENT(ERC,IT,KREK,GMP,PP)

ARGUMENTS: ERC I SCALAR ERROR OF CURRENT ITERATE
 IT I/O ITERATION COUNTER
 KREK I STEEPEST DESCENT RECTIFICATION NUMBER
 GMP I/O PREVIOUS GRADIENT MAGNITUDE (INPUT)
 CURRENT GRADIENT MAGNITUDE (OUTPUT)
 PP(3) I/O PREVIOUS GRADIENT (INPUT)
 CURRENT GRADIENT (OUTPUT)

SUBROUTINES SUPPORTED: TARGET

SUBROUTINES REQUIRED: TAROPT VMP

LOCAL SYMBOLS: ACK CURRENT ACCURACY LEVEL
 AER ABSOLUTE ERRORS OF TARGET PARAMETERS
 AUXN VALUES OF AUXILIARY PARAMETERS OF CURRENT
 ITERATE
 DD DIRECTIONAL DERIATIVE
 DELVM MAGNITUDE OF PREDICTED CORRECTION
 DEVI DEVIATION OF NOMINALLY-PREDICTED AUXILIARY
 PARAMETERS FROM CURRENT ITERATE VALUES
 DUMH DUMMY VARIABLES
 DUM DUMMY VARIABLES
 DVEE VELOCITY PERTURBATIONS
 DVM MAXIMUM ALLOWABLE VELOCITY INCREMENT
 ERB SCALAR ERROR OF NOMINALLY-PREDICTED STEP
 GC CURRENT GRADIENT
 GNC MAGNITUDE OF GC
 HB NOMINALLY PREDICTED STEP MAGNITUDE

HH CORRECTION MAGNITUDE AFTER CONSTRAINTS
 HS CORRECTION MAGNITUDE AFTER PARABOLIC FIT
 IEND FLAG SET TO 1 IF TOLERANCES ACCEPTABLE
 ON PERTURBED TRAJECTORY
 ISP2 SOI STOPPING CONDITION FLAG
 =0 DO NOT STOP AT SOI
 =1 STOP AT SOI
 PC DIRECTION OF CORRECTION
 PERR PERTURBED ERRORS
 PM MAGNITUDE OF UNNORMALIZED DIRECTION VECTOR
 QC UNIT VECTOR IN DIRECTION OF GRADIENT
 RSF FINAL STATE OF INTEGRATION

COMMON COMPUTED/USED:

DELTAV ISPH RIN TEN

COMMON COMPUTED:

ICL2 ICL INCMT RRF

COMMON USED:

AAUX	AC	ATAR	CTOL	DAUX
DELTAT	DTAR	DVMAX	D1	FAC
IPHASE	ISTOP	KUR	LEV	LVLS
NOPAR	PERV	TRTM	TWO	ZERO

DESENT Analysis

DESENT computes a correction to an initial velocity by the steepest descent or conjugate gradient techniques for use by TARGET.

The technique used is determined by the value of METHOD. DESENT takes n steps in the conjugate gradient directions before rectifying by making a steepest descent step where $n = \text{METHOD} - 1$. Thus if $\text{METHOD} = 1$, all steps are taken in the steepest descent direction.

Let the current iterate initial state be denoted \vec{r}, \vec{v} . Let the scalar error of the auxiliary parameters corresponding to this state be denoted ϵ . Let the perturbation size for the sensitivities be dv .

The current gradient \vec{g}_c is computed by numerical differencing. For the k -th component of \vec{g}_c the corresponding component of velocity is perturbed by dv

$$\vec{v}_p = \vec{v} + dv \begin{bmatrix} \delta_{1K}, \delta_{2K}, \delta_{3K} \end{bmatrix}^T \quad (1)$$

The initial state (\vec{r}, \vec{v}_p) is then propagated to the final stopping conditions. Let the auxiliary parameters of that trajectory be denoted $\vec{\alpha}_p$. The error associated with the perturbed state is then

$$\epsilon_p = \vec{W} \cdot (\vec{\alpha}_p - \vec{\alpha}^*) \quad (2)$$

where \vec{W} represents the weighting factors and $\vec{\alpha}^*$ are the desired target conditions. The k -th component of the current gradient is then

$$g_{cK} = \frac{\epsilon_p - \epsilon}{dv} \quad (3)$$

The corrected gradient is given by

$$\begin{aligned} \vec{p}_c &= \vec{g}_c && \text{steepest descent step} \\ &= \frac{|\vec{g}_c|^2}{|\vec{g}_p|^2} \vec{p}_p + \vec{g}_c && \text{conjugate gradient step} \end{aligned} \quad (4)$$

where the subscript c refers to a current parameter, p refers to a previous-step parameter.

The unit vector in the direction of the next step is then given by

$$\vec{q}_c = - \frac{\vec{p}_c}{p_c} \quad (5)$$

The directional derivative of the scalar error in the the direction \vec{q}_c is

$$d = \vec{g}_c \cdot \vec{q}_c \quad (6)$$

The nominal step size \bar{h} is computed from a linear approximation to null the error

$$\bar{h} = \frac{\epsilon}{-d} \quad (7)$$

The initial state corrected by this nominal correction is then propagated to the final stopping conditions and the resulting error $\bar{\epsilon}$ computed. The three conditions

$$\begin{aligned} y(o) &= \epsilon \\ y(\bar{h}) &= \bar{\epsilon} \\ y'(o) &= d \end{aligned} \quad (8)$$

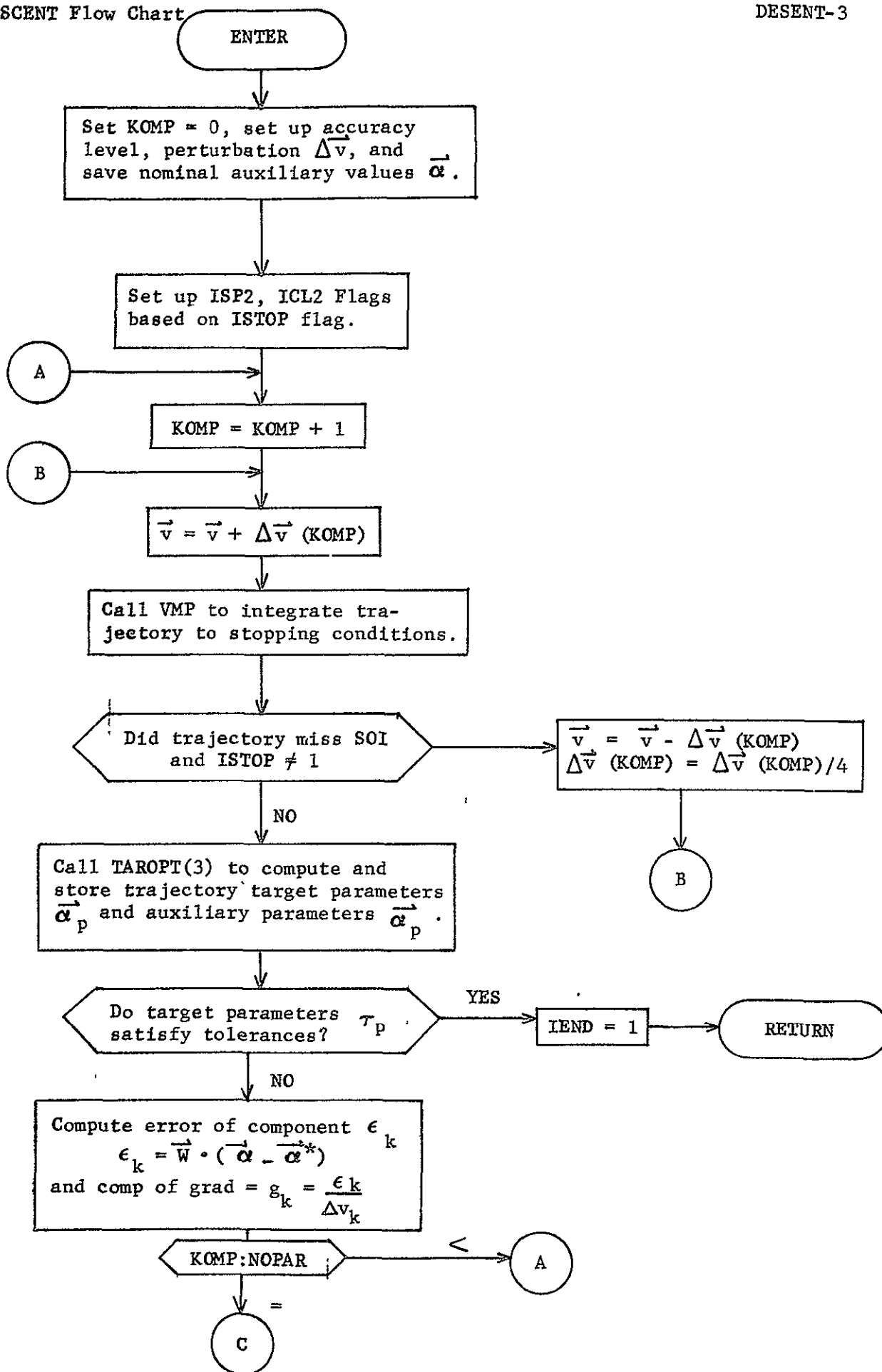
may now be applied to the formula of a parabola $y - \epsilon^* = a(x - h^*)^2$ to predict the optimal step size h^* yielding the minimum error ϵ^*

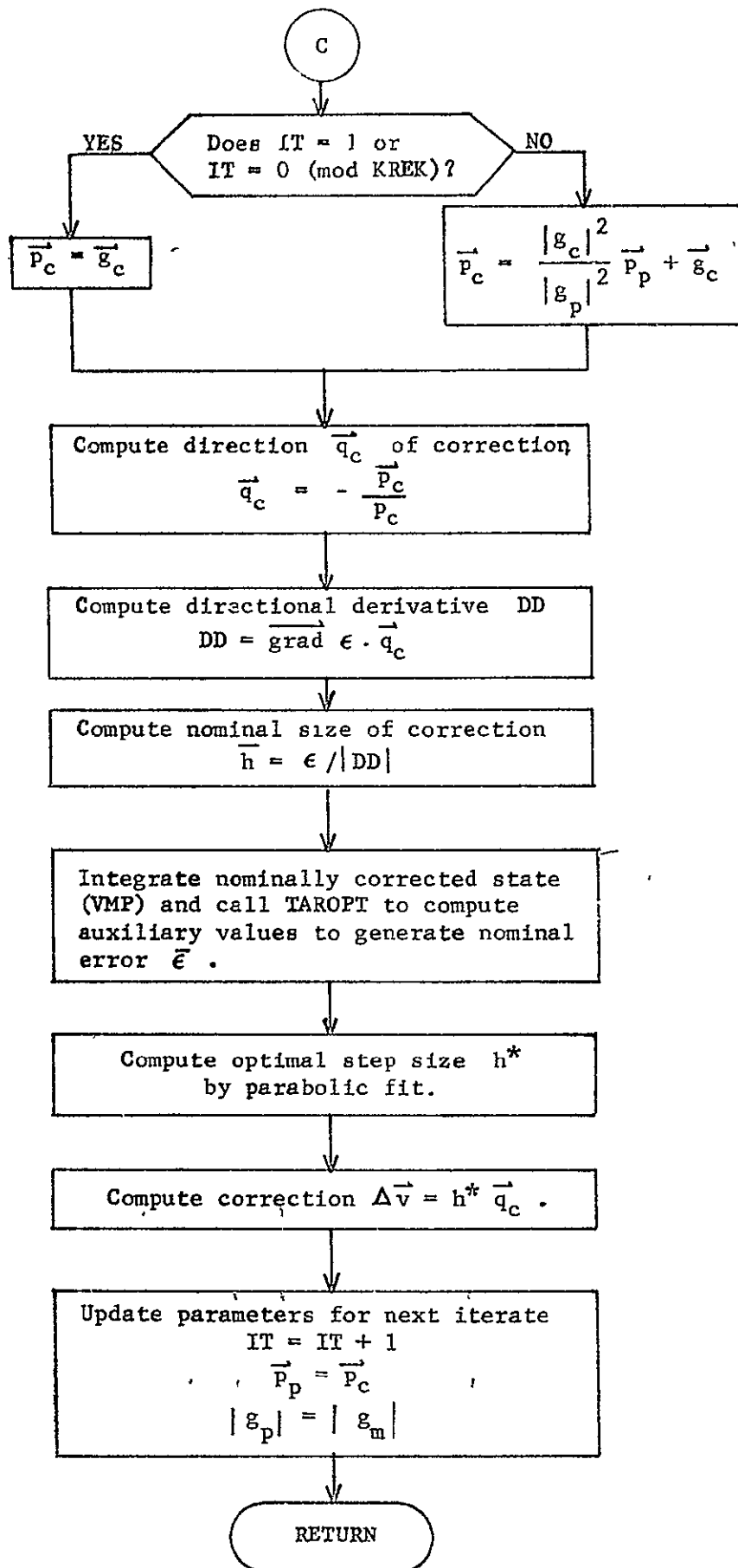
$$h^* = \frac{d\bar{h}^2}{2(d\bar{h} + \epsilon - \bar{\epsilon})} \quad (9)$$

The correction for the current is then given by

$$\Delta v = h^* \vec{q}_g \quad (10)$$

Reference: Myers, G. E., "Properties of the Conjugate Gradient and Davidson Methods", AAS Paper 68-081. Presented at 1968 AAS/AIAA Astrodynamics Specialist Conference, Jackson, Wyoming.





SUBROUTINE DIMPCP

PURPOSE: TO CALCULATE THE DESIRED IMPACT PLANE ASYMPTOTE PIERCE POINT COORDINATES GIVEN THE RIGHT ASCENSION AND DECLINATION OF A PROBE TARGET SITE

ARGUMENT:	A	I	SEMI-MAJOR AXIS OF APPROACH HYPERBOLA IN KM
	BM	O	MAGNITUDE OF B VECTOR IN KM
	DBR	O	DESIRED R.R IN KM
	DBT	O	DESIRED B.T IN KM
	DCP	I	DESIRED DECLINATION IN PROBE-SPHERE COORDINATES OF IMPACT SITE IN DEG
	RAP	I	DESIRED RIGHT ASCENSION IN PROBE-SPHERE COORDINATES OF IMPACT SITE IN DEG
	RPR	I	RADIUS OF THE PROBE SPHERE IN KM
	RV	I	PLANETOCENTRIC VECTOR IN DIRECTION OF CROSS PRODUCT OF SV BY ECLIPTIC POLE VECTOR
	SV	I	PLANETOCENTRIC VECTOR IN DIRECTION OF TRAJECTORY ASYMPTOTE
	TV	I	CROSS PRODUCT OF SV BY RV
	T	I	TRANSFORMATION MATRIX FROM ECLIPTIC TO PROBE-SPHERE COORDINATES

SUBROUTINES SUPPORTED: TPPROP TPRTRG

SUBROUTINES SUPPORTED:

LOCAL SYMBOLS:	BV	PLANETOCENTRIC UNIT VECTOR IN DIRECTION OF B VECTOR
	CSDCP	COSINE OF DECLINATION OF DESIRED IMPACT SITE
	CSOIF	COSINE OF THE ARC LENGTH BY WHICH DESIRED IMPACT SITE IS CLOSER TO SV THAN IS PERIAPSIS
	CSPHI	COSINE OF ANGLE BETWEEN SV AND VECTOR TO DESIRED IMPACT SITE
	CSRAP	COSINE OF RIGHT ASCENSION OF DESIRED IMPACT SITE

CSTHTS	COSINE OF TRUE ANAMOLY OF TRAJECTORY ASYMPTOTE
C1	COEFFICIENT USED IN CALCULATING REPOSITIONED IMPACT SITE
C2	COEFFICIENT USED IN CALCULATING REPOSITIONED IMPACT SITE
DCPSAV	SAVED VALUE OF IMPACT SITE DECLINATION IN DEG
E	ECCENTRICITY OF CONIC
FOUR	CONSTANT 4.
HALF	CONSTANT .5
ONE	CONSTANT 1.
RAPSAV	SAVED VALUE OF IMPACT SITE RIGHT ASCENSION IN DEG
RPVRV	PLANETOCENTRIC UNIT VECTOR TO DESIRED IMPACT SITE IN ECLIPTIC COORDINATES
RPV	PLANETOCENTRIC UNIT VECTOR TO DESIRED IMPACT SITE IN PROBE-SPHERE COORDINATES
RTD	CONVERSION FACTOR FROM RADIANS TO DEGREES
SNDOP	SINE OF DECLINATION OF DESIRED IMPACT SITE
SNPHI	SINE OF ANGLE BETWEEN SV AND VECTOR TO DESIRED IMPACT SITE
SNRAP	SINE OF RIGHT ASCENSION OF DESIRED IMPACT SITE
SNTHTS	SINE OF TRUE ANAMOLY OF TRAJECTORY ASYMPTOTE

DIMPCP Analysis

Subroutine DIMPCP converts the actual probe target parameters of declination δ and right ascension α of the trajectory impact point on the planet into the auxiliary target parameters of equivalent B·T and B·R. To do so it assumes the direction of the hyperbolic excess velocity and the energy of the trajectory are known so the \underline{S} , \underline{T} , and \underline{R}^* in the ecliptic frame and the semimajor axis, a , of the approach hyperbola are available as inputs. To complete the specification of the probe impact point, the subroutine also requires the radius, r , of the planet at impact as well as the transformation, L , from the inertial ecliptic frame to the coordinate system to which the right ascension and declination are referenced.

Derivation of the necessary equations is relatively straightforward once the appropriate variables are defined. Let $\underline{\rho}$ be a planet-centered unit vector in the direction of the impact point. Then, in the inertial ecliptic system,

$$\underline{\rho} = L^T \begin{pmatrix} \cos \alpha \cos \delta \\ \sin \alpha \cos \delta \\ \sin \delta \end{pmatrix} . \quad (1)$$

Define ϕ to be the unique angle on the closed interval from 0 to π between $\underline{\rho}$ and \underline{S} (see Fig. 1). Finally denote the true anomalies of $\underline{\rho}$ and \underline{S} by θ and θ_S , respectively.

First DIMPCP determines whether the desired impact point is indeed targetable. It is apparent from Figure 1 that

$$|\theta| = \phi - \theta_S \quad (2)$$

It is further obvious from the figure that the approach hyperbola will intersect the planet surface at true anomalies of both $+\theta$ and $-\theta$. Obviously only the negative true anomaly impact points are physically realizable since the trajectory stops at the first intersection with the planet. Hence DIMPCP requires the

$$\theta_S \leq \phi . \quad (3)$$

* For definitions of these vectors see the analysis section of the subroutine STIMP.

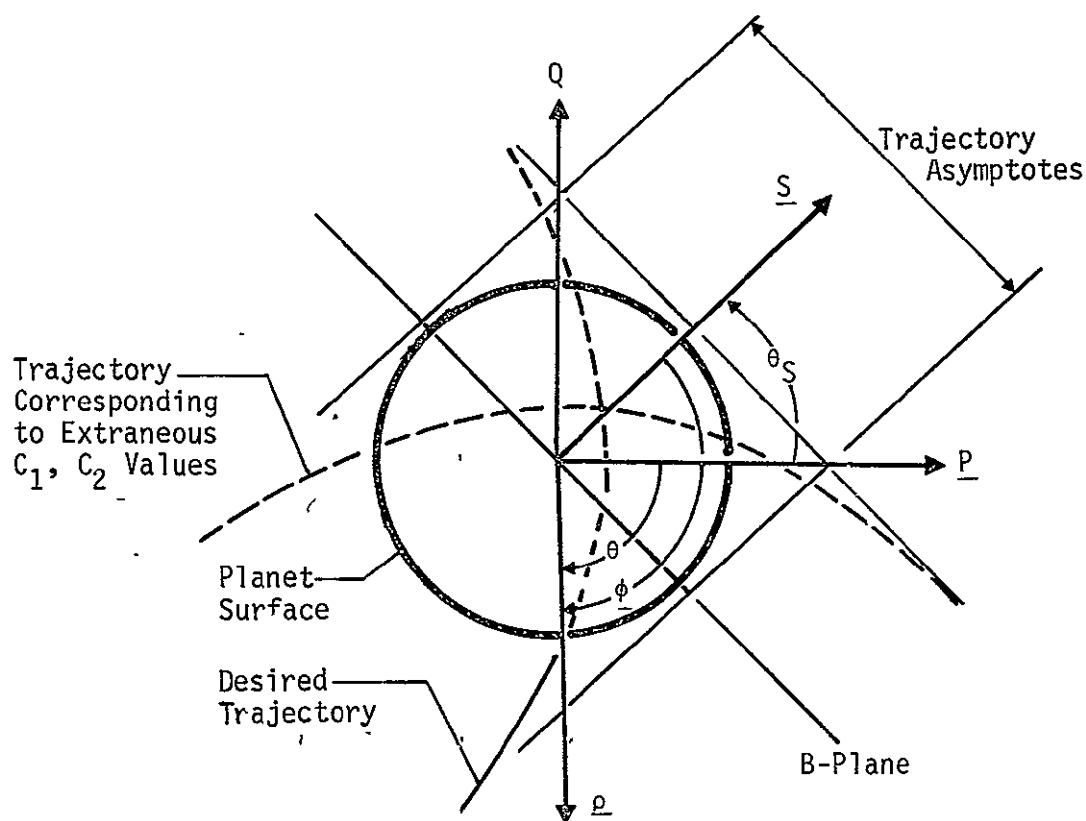


Figure 1 Geometry of Probe Impact

In other words, there is a circular region on the planet surface of radius θ_S about the outgoing pierce point of the \underline{S} vector inside of which no probes can be targeted. If $\underline{\rho}$ falls in this untargetable region, DIMPCP repositions the desired impact point direction to $\underline{\rho}'$, the nearest acceptable direction in the plane determined by \underline{S} and $\underline{\rho}$. Analytically this is done by expressing $\underline{\rho}'$ as a linear combination of $\underline{\rho}$ and \underline{S} ; that is

$$\underline{\rho}' = d_1 \underline{\rho} + d_2 \underline{S} \quad (4)$$

Then the constraints

$$||\underline{\rho}'|| = 1 \quad (5)$$

and

$$\underline{\rho} \cdot \underline{\rho}' = \cos (\theta_S - \phi) \quad (6)$$

are applied. These result in the following pair of simultaneous equations for d_1 and d_2 :

$$1 = d_1^2 + 2 d_1 d_2 \cos \phi + d_2^2 \quad (7)$$

$$\cos (\theta_S - \phi) = d_1 + d_2 \cos \phi \quad . \quad (8)$$

Solving (8) for d_1 in terms to d_2 and substituting into (7) produces the quadratic

$$1 - \cos^2 (\theta_S - \phi) = d_2^2 (1 - \cos^2 \phi) \quad . \quad (9)$$

Assuming $|\cos \phi| \neq 1$ leads then to the conclusion that

$$d_2 = \pm \sqrt{\frac{1 - \cos^2 (\theta_S - \phi)}{1 - \cos^2 \phi}} \quad . \quad (10)$$

Figure 2 geometrically interprets the two roots of equation (9) given by (10).

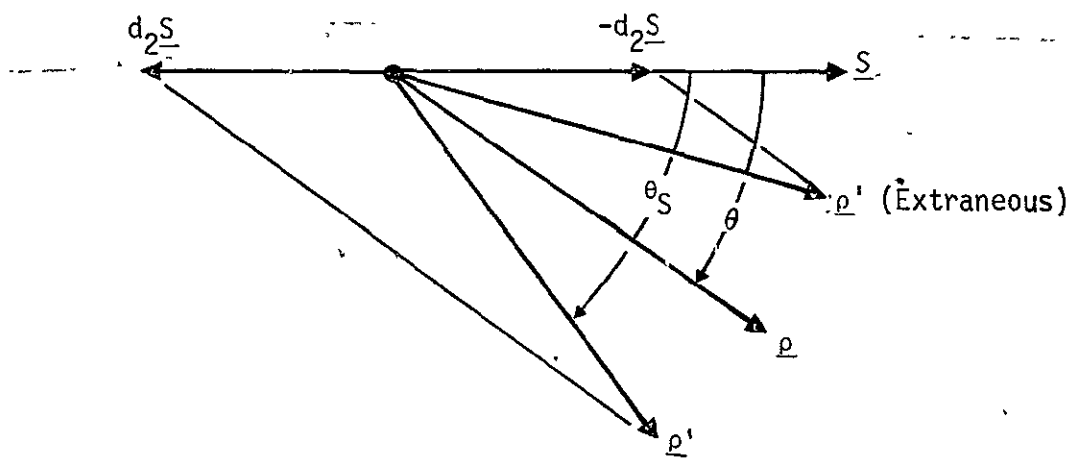


Figure 2 Geometrical Interpretation of the Two d_2 -Roots

Clearly the d_2 value corresponding to the positive radical is extraneous since it produces a $\underline{\rho}'$ nearer \underline{S} than $\underline{\rho}$. Hence

$$d_2 = - \sqrt{\frac{1 - \cos^2 (\theta_S - \phi)}{1 - \cos^2 \phi}} \quad (11)$$

$$d_1 = \cos (\theta_S - \phi) + \cos \phi \sqrt{\frac{1 - \cos^2 (\theta_S - \phi)}{1 - \cos^2 \phi}} \quad (12)$$

The exceptional case that $\cos \phi = -1$ cannot occur since then $\phi = \pi$ and hence $\theta_S < \phi$. However, $\cos \phi$ can equal 1. In this case $\underline{\rho}$ and \underline{S} are coincident so $\underline{\rho}'$ cannot be taken as a linear combination of the two. Further, no particular point on the boundary of the circular untargetable region recommends itself. Hence DIMPCP arbitrarily puts $\underline{\rho}'$ in the S-T plane as

$$\underline{\rho}' = \underline{S} \cos \theta_S + \underline{T} \sin \theta_S \quad (13)$$

On repositioning $\underline{\rho}$, DIMPCP prints out the right ascension α' and declination δ' of $\underline{\rho}'$ making use of the formulae

$$\alpha' = \tan^{-1} (\rho'_2 / \rho'_1) \quad (14)$$

$$\delta' = \sin^{-1} (\rho'_3) \quad (15)$$

Having repositioned $\underline{\rho}$, if necessary, DIMPCP calculates the magnitude of the desired \underline{B} . It can readily be shown that

$$\cos \theta_S = \frac{1}{e} \quad (16)$$

$$\sin \theta_S = \frac{\sqrt{e^2 - 1}}{e} \quad (17)$$

$$B = |a| \sqrt{e^2 - 1} \quad (18)$$

Recall the polar equation of the near-planet conic trajectory, namely

$$r = \frac{a(1 - e^2)}{1 + e \cos \theta} \quad (19)$$

Substituting equation (2) into (19) and rearranging gives

$$a(1 - e^2) = r [1 + e (\cos \phi \cos \theta_S + \sin \phi \sin \theta_S)] \quad (20)$$

Using equations (16) and (17) in (2) yields

$$a(1 - e^2) = r (1 + \cos \phi + \sqrt{e^2 - 1} \sin \phi) \quad (21)$$

Eliminating the eccentricity from equation (19) by means of (18) produces a quadratic in B , that is,

$$-B^2/a = r (1 + \cos \phi) - rB/a \quad (22)$$

Applying the quadratic formula to (20) gives

$$B = \frac{[r \sin \phi \pm \sqrt{r^2 \sin^2 \phi - 4 ar (1 + \cos \phi)}]}{2} \quad (23)$$

Since $1 + \cos \phi \geq 0$ for all ϕ and $a < 0$ for hyperbolic approach trajectories, the root corresponding to the negative radical in (21) produces a negative magnitude of B and hence must be extraneous. Thus

$$B = \frac{[r \sin \phi + \sqrt{r^2 \sin^2 \phi - 4 ar (1 + \cos \phi)}]}{2} \quad (24)$$

One can further conclude from the radicand of (23) that a solution for B will exist if, and only if,

$$4 ar (1 + \cos \theta) \leq r^2 (1 - \cos^2 \phi) \quad (25)$$

or equivalently if $\cos \phi \neq -1$

$$\cos \phi \leq 1 - \frac{4a}{r} \quad (26)$$

Since a is negative for a hyperbolic approach, this last inequality is always true. Further if $\cos \phi = -1$, $B = 0$. Hence equation (22) always has the unique nonnegative solution given by (24).

Next DIMPCP computes the direction of \underline{B} . Since the desired \underline{B} must lie in the plane determined by \underline{S} and $\underline{\rho}$, there must exist real numbers C_1 and C_2 so that

$$\underline{B}/B = C_1 \underline{S} + C_2 \underline{\rho} \quad . \quad (27)$$

Applying the constraints that $|\underline{B}/B| = 1$ and $\underline{B} \cdot \underline{S} = 0$ respectively

$$C_1^2 + 2 C_1 C_2 \cos \phi + C_2^2 = 1 \quad (28)$$

$$C_1 + C_2 \cos \phi = 0 \quad . \quad (29)$$

Solving these two equations simultaneously for a and b gives

$$C_2 = \pm 1/\sin \phi \quad (30)$$

$$C_1 = \mp \cot \phi \quad . \quad (31)$$

The negative C_2 root and the corresponding positive C_1 root are extraneous since they place \underline{B} on the side of \underline{S} opposite to $\underline{\rho}$ as shown in Figure 1. Substituting the correct pair of roots from (30) and (31) into (27) gives the direction of the desired \underline{B} as

$$\underline{B}/B = (\underline{\rho} - \underline{S} \cos \phi)/\sin \phi \quad . \quad (32)$$

Clearly in the exceptional case that $\sin \phi = 0$, the trajectory passes through the center of the planet coinciding with its asymptote so that $\underline{B} = \underline{0}$.

Finally DIMPCP calculates the desired $B \cdot T$ and $B \cdot R$ coordinates now that \underline{B} is known:

$$B \cdot T = B_1 T_1 + B_2 T_2 \quad (33)$$

$$B \cdot R = B_1 R_1 + B_2 R_2 + B_3 R_3 \quad . \quad (34)$$

FUNCTION DINCOS

PURPOSE: TO CALCULATE THE INVERSE COSINE IN DEGREES OF GIVEN
 REAL ARGUMENT SETTING RESULT TO 0. (180.) IF ARGUMENT
 IS GREATER THAN 1. (LESS THAN -1.)

ARGUMENT: X I REAL NUMBER WHOSE INVERSE COSINE IS TO BE
 EVALUATED

SUBROUTINES SUPPORTED: CAREL TPPROP

LOCAL SYMBOLS: DINCOS INVERSE COSINE IN DEGREES OF ARGUMENT X

 ONE CONSTANT 1.

 RTD CONVERSION FACTOR FROM RADIANS TO DEGREES

DINSIN-A

FUNCTION DINSIN

PURPOSE: TO CALCULATE THE INVERSE SINE IN DEGREES OF GIVEN REAL
 ARGUMENT SETTING RESULT TO 90. (-90.) IF ARGUMENT
 IS GREATER THAN 1. (LESS THAN -1.)

ARGUMENT: X REAL NUMBER WHOSE INVERSE SINE IS TO BE
 EVALUATED

SUBROUTINES SUPPORTED: DIMPCP IMPCT TPPROP TPRTRG

LOCAL SYMBOLS: DINSIN INVERSE SINE IN DEGREES OF ARGUMENT X

 ONE CONSTANT 1.

 RTD CONVERSION FACTOR FROM RADIANS TO DEGREES

SUBROUTINE DYNO

PURPOSE: COMPUTE ASSUMED AND ACTUAL DYNAMIC NOISE COVARIANCE
MATRIX IN THE ERROR ANALYSIS PROGRAM

CALLING SEQUENCE: CALL DYNO(ICODE)

ARGUMENT: ICODE I =0 ASSUMED
 =1 ACTUAL

SUBROUTINES SUPPORTED: ERRANN SETEVN GUIDM PRED

LOCAL SYMBOLS: D2 SQUARE OF (DELT*TM)

COMMON COMPUTED: Q QPR

COMMON USED: DELTM DNGN IDNF TM
 IGDNF GDNEN

DYNØ Analysis

Subroutine DYNØ evaluates the assumed dynamic covariance matrix Q over the time interval $t = t_{k+1} - t_k$ if ICØDE = 0. If ICØDE = 1 the actual dynamic noise covariance matrix Q' is evaluated over the same interval. In either case the dynamic noise covariance matrix is assumed to have the form

$$Q = \text{diag} \left(\frac{1}{4} K_1 \Delta t^4, \frac{1}{4} K_2 \Delta t^4, \frac{1}{4} K_3 \Delta t^4, K_1 \Delta t^2, K_2 \Delta t^2, K_3 \Delta t^2 \right)$$

where dynamic noise constants K_1 , K_2 , and K_3 have units of km^2/s^4 . To compute the actual dynamic noise covariance matrix Q' , we simply replace K_1 , K_2 , and K_3 with the actual dynamic noise constants K_1' , K_2' , and K_3' , respectively.

SUBROUTINE DYNOS

PURPOSE: COMPUTE DYNAMIC NOISE COVARIANCE MATRIX AND THE ACTUAL DYNAMIC NOISE (UNMODELED ACCELERATION) IN THE SIMULATION PROGRAM

CALLING SEQUENCE: CALL DYNOS(ICODE)

ARGUMENT: ICODE I INTERNAL CODE TO DETERMINE IF THE DYNAMIC NOISE MATRIX IS COMPUTED OR IF THE ACTUAL DYNAMIC NOISE IS CALCULATED

SUBROUTINES SUPPORTED: SIMULL SETEVS GUISIM PRESIM

LOCAL SYMBOLS: DT INTERNAL TIME INCREMENT
 D2 SQUARE OF (DELTM*TM)
 IC INTERNAL CODE ON DT CALCULATION
 T1 CURRENT TIME
 T2 CURRENT TIME + DELTA TIME

COMMON COMPUTED/USED: W

COMMON COMPUTED: Q

COMMON USED: DELTM DNCN HALF IDNF TM
 TRTM1 TTIM1 TTIM2 UNMAC ZERO

DYNOS Analysis

Subroutine DYNOS performs two functions. It's first function is identical to that of subroutine DYNØ, namely, to evaluate the dynamic noise covariance matrix Q over the time interval $\Delta t = t_{k+1} - t_k$.

The second function of subroutine DYNØS is to compute the actual dynamic noise $\vec{\omega}_{k+1}$, which represents the integrated effect of unmodelled accelerations acting on the spacecraft over the time interval Δt . Actual dynamic noise $\vec{\omega}_{k+1}$ is used elsewhere in the program to compute the actual state deviations of the spacecraft from the most recent nominal trajectory.

If we define $\vec{\omega}_{k+1} = \begin{bmatrix} \vec{\omega}_{r_{k+1}}, & \vec{\omega}_{v_{k+1}} \end{bmatrix}^T$, where

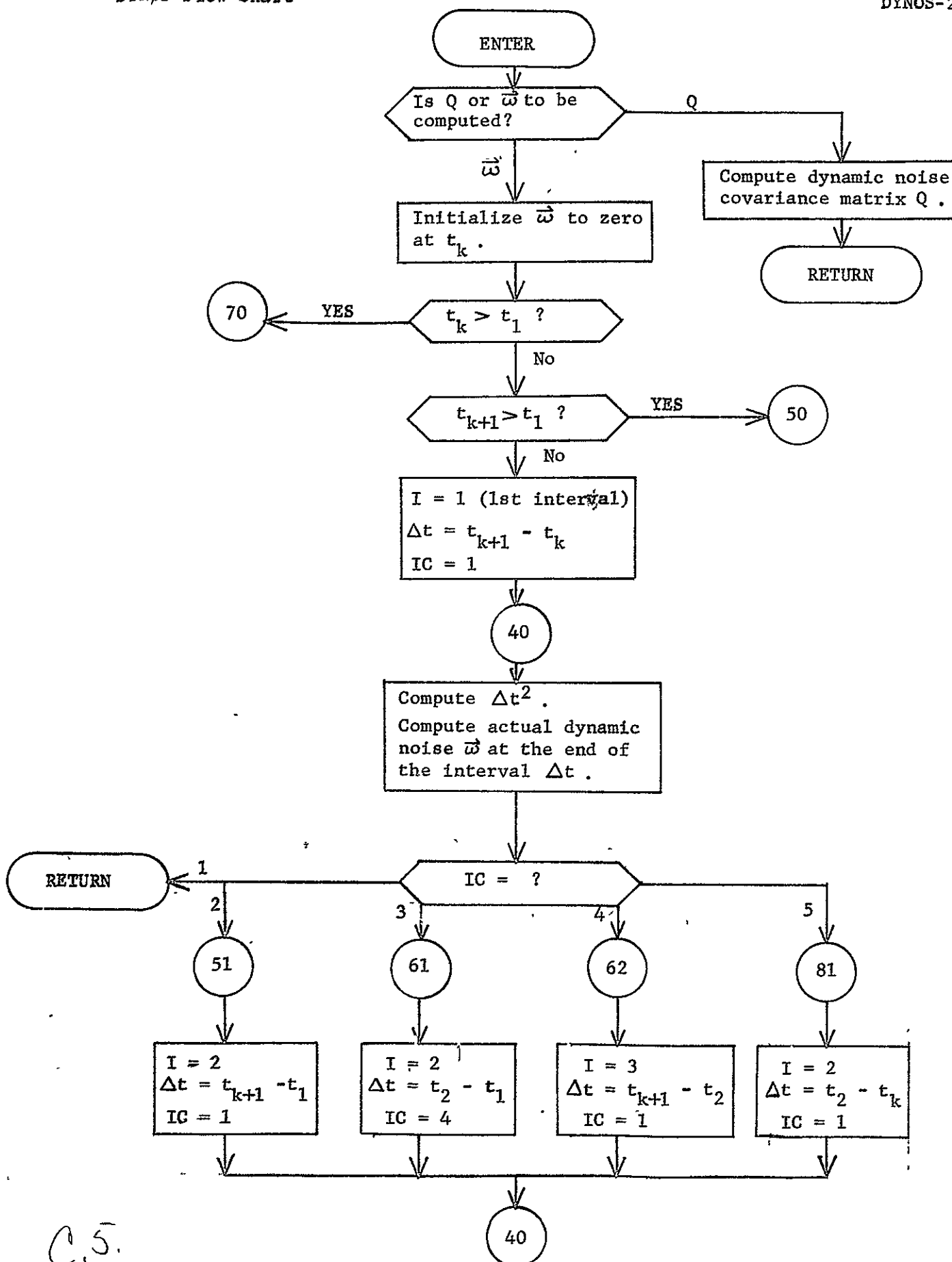
$\vec{\omega}_{r_{k+1}}$ and $\vec{\omega}_{v_{k+1}}$ denote the contributions of unmodelled

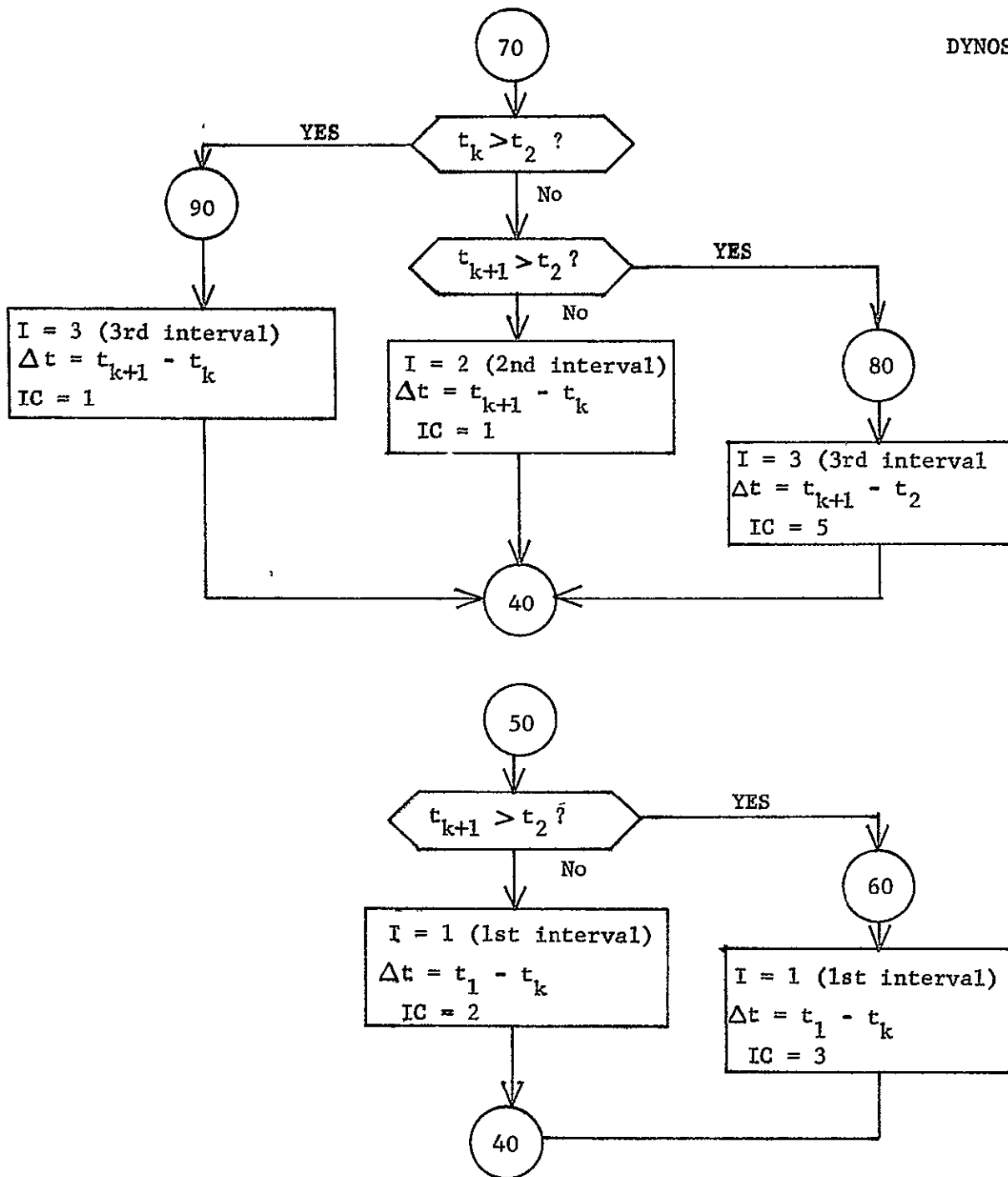
accelerations to spacecraft position and velocity, respectively, and if we assume constant unmodelled acceleration \vec{a} , then

$$\vec{\omega}_{r_{k+1}} = \frac{\vec{a}}{2} (t_{k+1} - t_k)^2 + \vec{\omega}_{v_k} (t_{k+1} - t_k) + \vec{\omega}_{r_k}$$

$$\vec{\omega}_{v_{k+1}} = \vec{a} (t_{k+1} - t_k) + \vec{\omega}_{v_k}$$

The program permits the entire trajectory to be divided into three arbitrary consecutive intervals, over each of which a different constant unmodelled acceleration \vec{a} can be specified. These intervals are represented by (t_0, t_1) , (t_1, t_2) , and (t_2, t_f) , where t_0 is the initial trajectory time and t_f is the final trajectory time. If t_k and t_{k+1} occur in different intervals, then the above equations must be evaluated piece-wise over (t_k, t_{k+1}) .





SUBROUTINE EIGHY

PURPOSE: TO CONTROL THE COMPUTATION OF EIGENVALUES, EIGENVECTORS,
AND HYPERELLIPSOIDS.

CALLING SEQUENCE: CALL EIGHY(VEIG,FOX,HARG,IFMT)

ARGUMENT: VEIG I MATRIX TO BE DIAGONALIZED
FOX I FINAL OFF-DIAGONAL ANNIHILATION VALUE
HARG I MATRIX FOR WHICH THE HYPERELLIPSOID IS TO
BE COMPUTED
IFMT I FORMAT FLAG
=1, PRINT POSITION EIGENVALUE TITLE
=2, PRINT VELOCITY EIGENVALUE TITLE
=3, PRINT EIGENVALUE TITLE

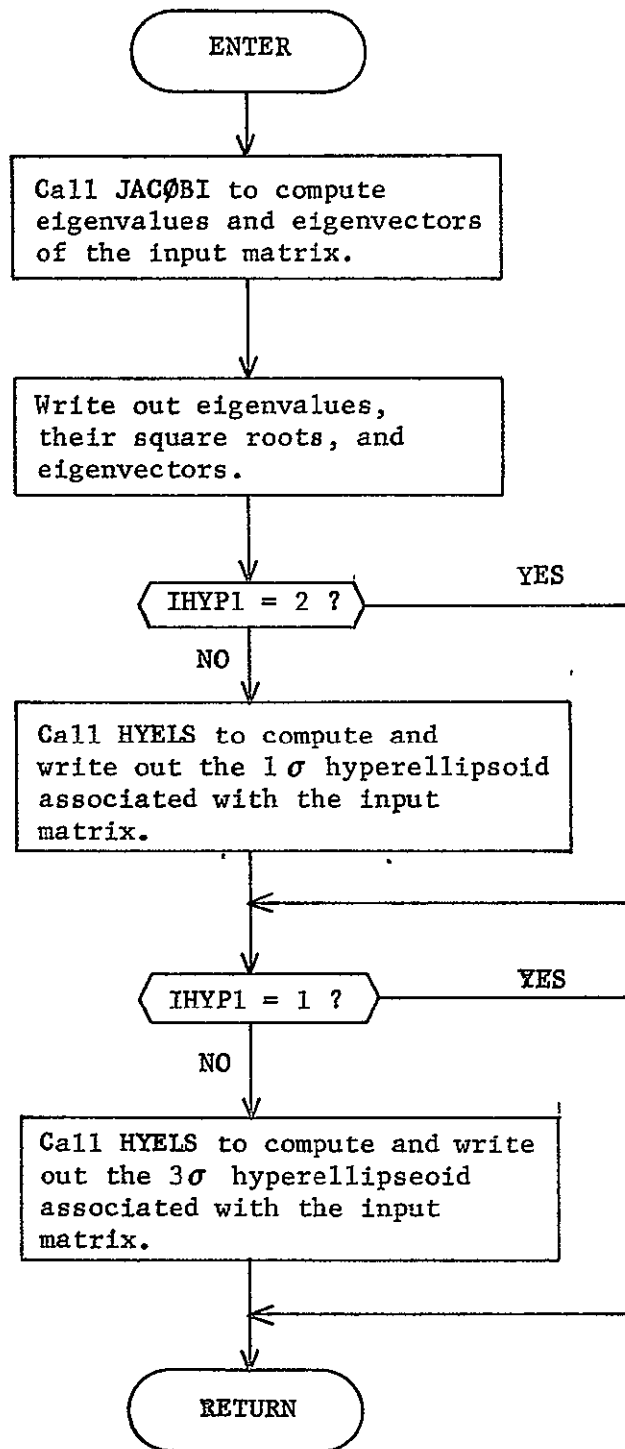
SUBROUTINES SUPPORTED: SETEVS GUISIM GUISS PRESIM GUIDM

SUBROUTINES REQUIRED: HYELS JACOBI

LOCAL SYMBOLS: EGVCT EIGENVECTOR MATRIX
EGVL EIGENVALUE MATRIX
OUT SQUARE ROOTS OF EIGENVALUES

COMMON USED: IHYP1

EIGHY Flow Chart



SUBROUTINE ELCAR

PURPOSE: TRANSFORMATION OF CONIC ELEMENTS TO CARTESIAN COORDINATES

CALLING SEQUENCE: CALL ELCAR(GM,A,E,W,XI,XN,TA,R,RH,V,VM,TFP)

ARGUMENTS	GM	I	GRAVITATIONAL CONSTANT OF CENTRAL BODY
	A	I	SEMI-MAJOR AXIS
	E	I	ECCENTRICITY
	W	I	ARGUMENT OF PERIAPSIS
	XI	I	INCLINATION IN REFERENCE SYSTEM
	XN	I	LONGITUDE OF ASCENDING NODE
	TA	I	TRUE ANOMALY
	R(3)	O	POSITION VECTOR IN REFERENCE SYSTEM
	RH	O	POSITION MAGNITUDE
	V(3)	O	VELOCITY VECTOR IN REFERENCE SYSTEM
	VM	O	VELOCITY MAGNITUDE
	TFP	O	TIME FROM PERIAPSIS

SUBROUTINES SUPPORTED: DATAS VMP NONLIN COPINS NONINS
DATA HELIO MULTAR CPROP

SUBROUTINES REQUIRED: NONE

LOCAL SYMBOLS:	AUXF	ECCENTRIC ANOMALY (HYPERBOLIC CASE)
	AVA	MEAN ANOMALY (ELLIPTIC CASE)
	CI	COSINE OF INCLINATION
	CK	VELOCITY FACTOR USED TO CALCULATE FINAL VELOCITY VECTOR
	CN	COSINE OF LONGITUDE OF ASCENDING NODE
	COSEA	COSINE OF ECCENTRIC ANOMALY (ELLIPTIC CASE)
	CT	COSINE OF TRUE ANOMALY

CW	COSINE OF SUM OF ARGUMENT OF PERIAPSIS AND TRUE ANOMALY/ COSINE OF ARGUMENT OF PERIAPSIS
DIV	THE SUM $1.+E*(COS(TA/RAD))$. USED AS A DIVISOR IN SUBSEQUENT EQUATIONS TO CALCULATE TFP
EA	ECCENTRIC ANOMALY (ELLIPTIC CASE)
P	SEMI-LATUS RECTUM
RAD	DEGREES TO RADIANS CONVERSION FACTOR
SINEA	SINE OF ECCENTRIC ANOMALY (ELLIPTIC CASE)
SINHF	HYPERBOLIC SINE OF AUXF
SI	SINE OF INCLINATION
SN	SINE OF LONGITUDE OF ASCENDING NODE
ST	SINE OF TRUE ANOMALY
SW	SINE OF THE SUM OF ARGUMENT OF PERIAPSIS AND TRUE ANOMALY/ SINE OF ARGUMENT OF PERIAPSIS
TANG	INTERMEDIATE VARIABLE USED TO CALCULATE SINHF

ELCAR Analysis

ELCAR transforms the standard conic elements of a massless point referenced to a gravitational body to cartesian position and velocity components with respect to that body.

Let the gravitational constant of the body be denoted μ and the given conic elements $(a, e, i, \omega, \Omega, f)$. The semilatus rectum p is

$$p = a (1 - e^2) \quad (1)$$

Then the magnitude of the radius vector is given by

$$r = \frac{p}{1 + e \cos f} \quad (2)$$

The unit vector in the direction of the position vector is

$$\begin{aligned} u_x &= \cos(\omega + f) \cos \Omega - \cos i \sin(\omega + f) \sin \Omega \\ u_y &= \cos(\omega + f) \sin \Omega + \cos i \sin(\omega + f) \cos \Omega \\ u_z &= \sin(\omega + f) \sin i \end{aligned} \quad (3)$$

The position vector \vec{r} is therefore

$$\vec{r} = r \hat{u} \quad (4)$$

The velocity vector \vec{v} is given by

$$\begin{aligned} v_x &= \sqrt{\frac{\mu}{p}} \left[(e + \cos f) (-\sin \omega \cos \Omega - \cos i \sin \Omega \cos \omega) \right. \\ &\quad \left. - \sin f (\cos \omega \cos \Omega - \cos i \sin \Omega \sin \omega) \right] \\ v_y &= \sqrt{\frac{\mu}{p}} \left[(e + \cos f) (-\sin \omega \sin \Omega + \cos i \cos \Omega \cos \omega) \right. \\ &\quad \left. - \sin f (\cos \omega \sin \Omega + \cos i \cos \Omega \sin \omega) \right] \\ v_z &= \sqrt{\frac{\mu}{p}} \left[(e + \cos f) \sin i \cos \omega - \sin f \sin i \sin \omega \right] \end{aligned} \quad (5)$$

The conic time from periapsis t_p is computed from different formulae depending upon the sign of the semi-major axis. For $a > 0$ (elliptical motion)

$$t_p = \sqrt{\frac{a^3}{\mu}} (E - e \sin E)$$

$$\cos E = \frac{e + \cos f}{1 + e \cos f} \quad \sin E = \frac{\sqrt{1 - e^2} \sin f}{1 + e \cos f} \quad (6)$$

For $a < 0$ (hyperbolic motion) the time from periapsis is

$$t_p = \sqrt{\frac{a^3}{\mu}} (e \sinh H - H)$$

$$\tanh \frac{H}{2} = \sqrt{\frac{e - 1}{e + 1}} \tan \frac{f}{2} \quad (7)$$

SUBROUTINE ELIPT

PURPOSE: TO CALCULATE TIME FROM PERIAPSIS ON AN ELLIPSE GIVEN
TRUE ANAMOLY

ARGUMENT: A I SEMI-MAJOR AXIS OF ELLIPSE IN KM
E I ECCENTRICITY OF ELLIPSE
ORBE I RECIPROCAL OF MEAN ORBITAL RATE IN
SEC/RAD
R I RADIUS IN KM
SNTA I SINE OF TRUE ANAMOLY
TIME O TIME FROM PERIAPSIS IN SEC

SUBROUTINES SUPPORTED: CAREL

LOCAL SYMBOLS: CSE COSINE OF ECCENTRIC ANAMOLY
SNE SINE OF ECCENTRIC ANAMOLY
TWOPI CONSTANT $2 \cdot \pi$

SUBROUTINE EPHEM

PURPOSE: TO COMPUTE THE CARTESIAN STATE OF DESIRED BODIES AT SPECIFIED TIMES ACCORDING TO TWO OPTIONS:
 (1) ECLIPTIC COORDINATES OF ONE BODY RELATIVE TO ITS REFERENCE BODY (SUN FOR PLANETS, EARTH FOR MOON)
 (2) ECLIPTIC COORDINATES OF ALL GRAVITATIONAL BODIES RELATIVE TO THE INERTIAL COORDINATE SYSTEM (EITHER HELIOCENTRIC OR BARYCENTRIC).

CALLING SEQUENCE: CALL EPHEM(IC,D,N)

ARGUMENT: D I JULIAN DATE OF REFERENCE TIME (REFERENCED 1950)
 IC I FLAG SET EQUAL TO 1 FOR OPTION 1 AND TO 0 FOR OPTION 2
 N I NUMBER OF GRAVITATIONAL BODIES TO BE COMPUTED

SUBROUTINES SUPPORTED: HELIO LAUNCH LUNTAR MULCON MULTAR
 EXCUTE TRAPAR VMP DATAS PCTM
 PRINT4 PSIM TRAKS GUISIM GUISS
 PRNTS4 DATA PRINT3 TRAKM GUIDM
 GUID

SUBROUTINES REQUIRED: CENTER

LOCAL SYMBOLS: A SEMI-MAJOR AXIS OF LUNAR CONIC
 DD ONE TEN-THOUSANDTH TIMES THE INPUT ARGUMENT D FOR COMPUTATIONS IN FN1, FN2
 E ECCENTRICITY OF LUNAR CONIC
 ECAM ECCENTRIC ANOMALY USED TO SOLVE KEPLER EQUATION
 ECC ECCENTRICITY USED TO SOLVE KEPLER EQUATION
 EM MEAN ANOMALY OF LUNAR CONIC
 E2 E SQUARED
 E3 E CUBED
 FCTR VELOCITY DIVIDED BY RADIUS
 FN1 STATEMENT FUNCTION DEFINING A THIRD ORDER POLYNOMIAL. USED IN COMPUTATION OF MEAN ANOMALY OF INNER PLANETS AND OF MOON

FN2 STATEMENT FUNCTION DEFINING A FIRST ORDER
 POLYNOMIAL. USED IN MEAN ANOMALY COMPUTA-
 TIONS OF THE OUTER PLANETS

I INDEX FOR LOGIC CONTROL

IJKL INCREMENT COUNTER IN SOLUTION OF KEPLER
 EQUATION

IN INDEX, ROW OF F OF LUNAR COORDINATES

IND INDEX, ROW OF F OF COORDINATES OF THE
 I-TH PLANET

ITEMP INTERMEDIATE VARIABLE USED TO NORMALIZE
 CONIC ANGLES

ITEST INTERNAL CODE WHICH DETERMINES IF
 COORDINATES OF EARTH ARE BEING CALCULATED
 IN ORDER TO COMPUTE THOSE OF MOON

ITEST2 INTERNAL CODE WHICH DETERMINES IF
 COORDINATES OF EARTH HAVE BEEN COMPUTED
 PRIOR TO COMPUTING THOSE OF THE MOON

K INDEX USED IN CALCULATION OF MEAN ANOMALY

P SEMI-LATUS RECTUM

PI2 TWO TIMES THE MATHEMATICAL CONSTANT PI

R HELIOCENTRIC RADIUS OF PLANET

TRG ARRAY OF TRIGONOMETRIC FUNCTIONS OF
 SPECIFIED ANGLES

VEL VELOCITY OF PLANET

WX X-COMPONENT OF INTERMEDIATE VECTOR, W

WY Y-COMPONENT OF INTERMEDIATE VECTOR, W

WZ Z-COMPONENT OF INTERMEDIATE VECTOR, W

COMMON COMPUTED/USED:

ELMNT	F	T	XP
-------	---	---	----

COMMON USED:

CN	EMN	IBARY	NBODYI	NO
ONE	PMASS	ST	TWOPI	TWO
ZERO				

EPHEM Analysis

EPHEM first determines the current value for the mean anomaly of the pertinent body. The mean anomaly M is computed from

$$\begin{aligned} M &= M_0 + M_1 t + M_2 t^2 + M_3 t^3 && \text{for inner planets} \\ M &= M_0 + M_1 t && \text{for outer planets} \\ M &= L_0 + L_1 t + L_2 t^2 + L_3 t^3 - \tilde{\omega}(t) && \text{for the moon} \end{aligned}$$

Kepler's equation $M = E - e \sin E$ is then solved iteratively to determine the eccentric anomaly E . The subsequent computations are basic conic manipulations:

$$p = a(1 - e^2)$$

$$r = a(1 - e \cos E)$$

$$v = \sqrt{\mu \left(\frac{2}{r} - \frac{1}{a} \right)}$$

$$\cos f = \frac{p - r}{er}$$

$$\sin f = \sqrt{1 - \cos^2 f} \operatorname{sgn}(\sin E)$$

$$\cos \gamma = \frac{\sqrt{\mu p}}{rv}$$

$$\sin \gamma = \sqrt{1 - \cos^2 \gamma} \operatorname{sgn}(\sin E)$$

$$\omega = \tilde{\omega} - Q$$

The cartesian position and velocity relative to the reference body are then

$$\vec{r} = r_x \hat{i} + r_y \hat{j} + r_z \hat{k}$$

$$r_x = r \cos(\omega + f) \cos \Omega - r \sin(\omega + f) \sin \Omega \cos i$$

$$r_y = r \cos(\omega + f) \sin \Omega + r \sin(\omega + f) \cos \Omega \cos i$$

$$r_z = r \sin(\omega + f) \sin i$$

$$\vec{v} = \frac{v}{r} \left[(\hat{w} \times \vec{r}) \cos \gamma + \vec{r} \sin \gamma \right]$$

$$\text{where } \hat{w} = (\sin i \sin \Omega) \hat{i} - (\sin i \cos \Omega) \hat{j} + (\cos i) \hat{k}$$

When option 1 is used, the reference body for all the planets is the sun while the reference body for the moon is the earth.

When option 2 is used with heliocentric inertial coordinates, the cartesian state of the earth is added to the cartesian state of the moon to convert the state of the moon to heliocentric coordinates before storing that state in the F-array.

When option 2 is used with barycentric inertial coordinates, subroutine CENTER is called to convert all elements to barycentric coordinates before storing in the F-array.

PROGRAM ERRANN

PURPOSE: TO CONTROL THE COMPUTATIONAL FLOW THROUGH THE BASIC
CYCLE (MEASUREMENT PROCESSING) AND ALL EVENTS IN THE
ERROR ANALYSIS MODE.

SUBROUTINES SUPPORTED: ERRON

SUBROUTINES REQUIRED:	SCHED	NTM	PSIM	DYNO	TRAKM
	MENO	GNAVM	PRINT3	SETEVN	GUIDM
	MEAN	GPRINT			

LOCAL SYMBOLS:	AY	DUMMY VARIABLE
	ICL2S	TEMPORARY STORAGE FOR ICL2
	ICODE	EVENT CODE
	IPRN	MEASUREMENT COUNTER FOR PRINTING
	ISP2S	TEMPORARY STORAGE FOR ISP2
	NEVENT	EVENT COUNTER
	SPHERS	TEMPORARY STORAGE FOR SPHERE(NTP)
	TRTM2	TIME OF THE MEASUREMENT

COMMON COMPUTED/USED:	ICODE	MCNTR	RI	TEVN	TRTM1
	XF	XI			

COMMON COMPUTED:	DELT	TLAST	SPHERE	ICL2	ISP2
	XPHI	ALFA	DELT	ABW	

COMMON USED:	FNTM	IEVNT	IPRINT	ISTMC	NEV
	NMN	NR	NTMC	RF	TEV
	UCNTRL	IUTC	KKWIT	NOGEN	T7

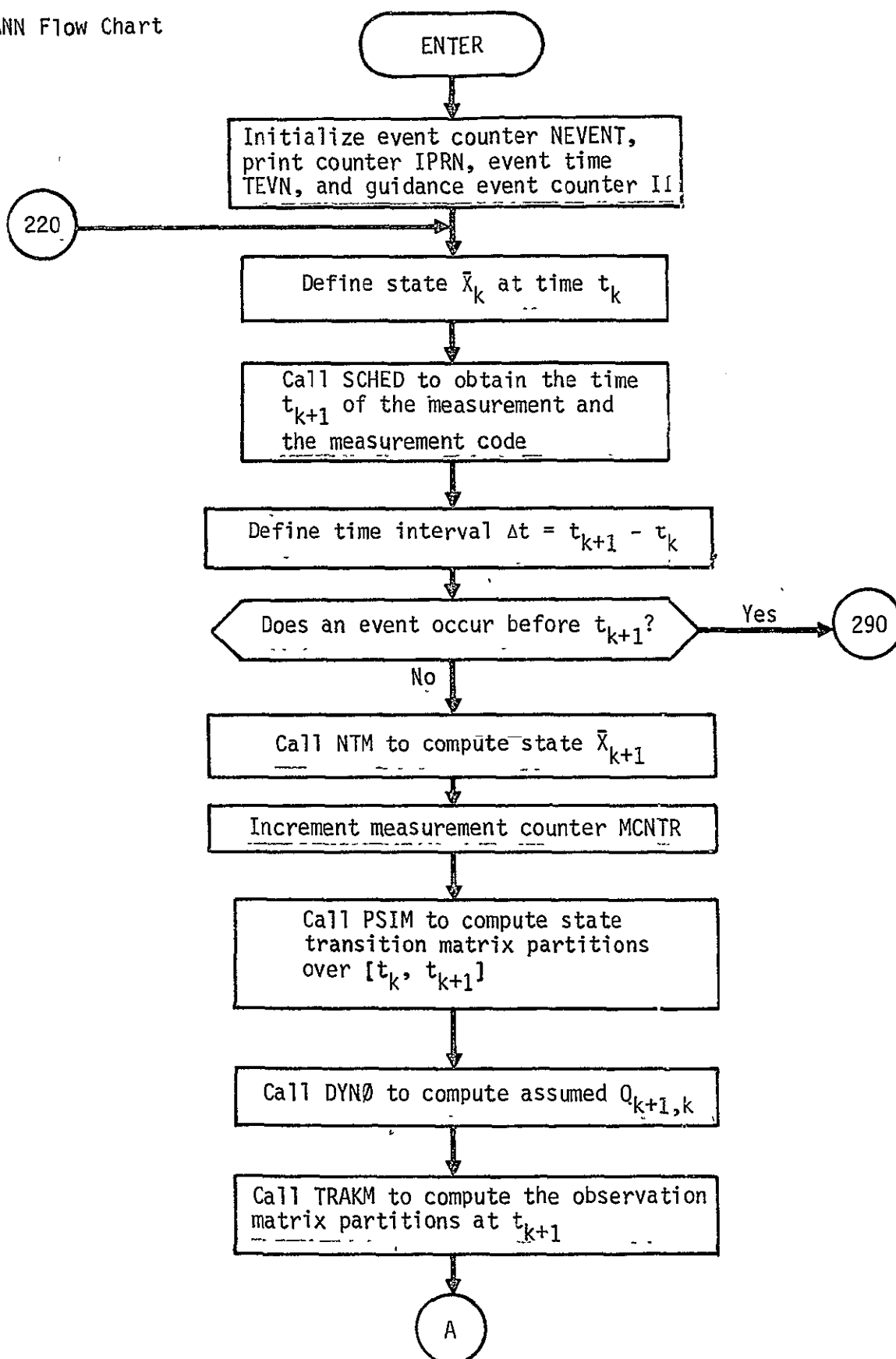
ERRANN Analysis

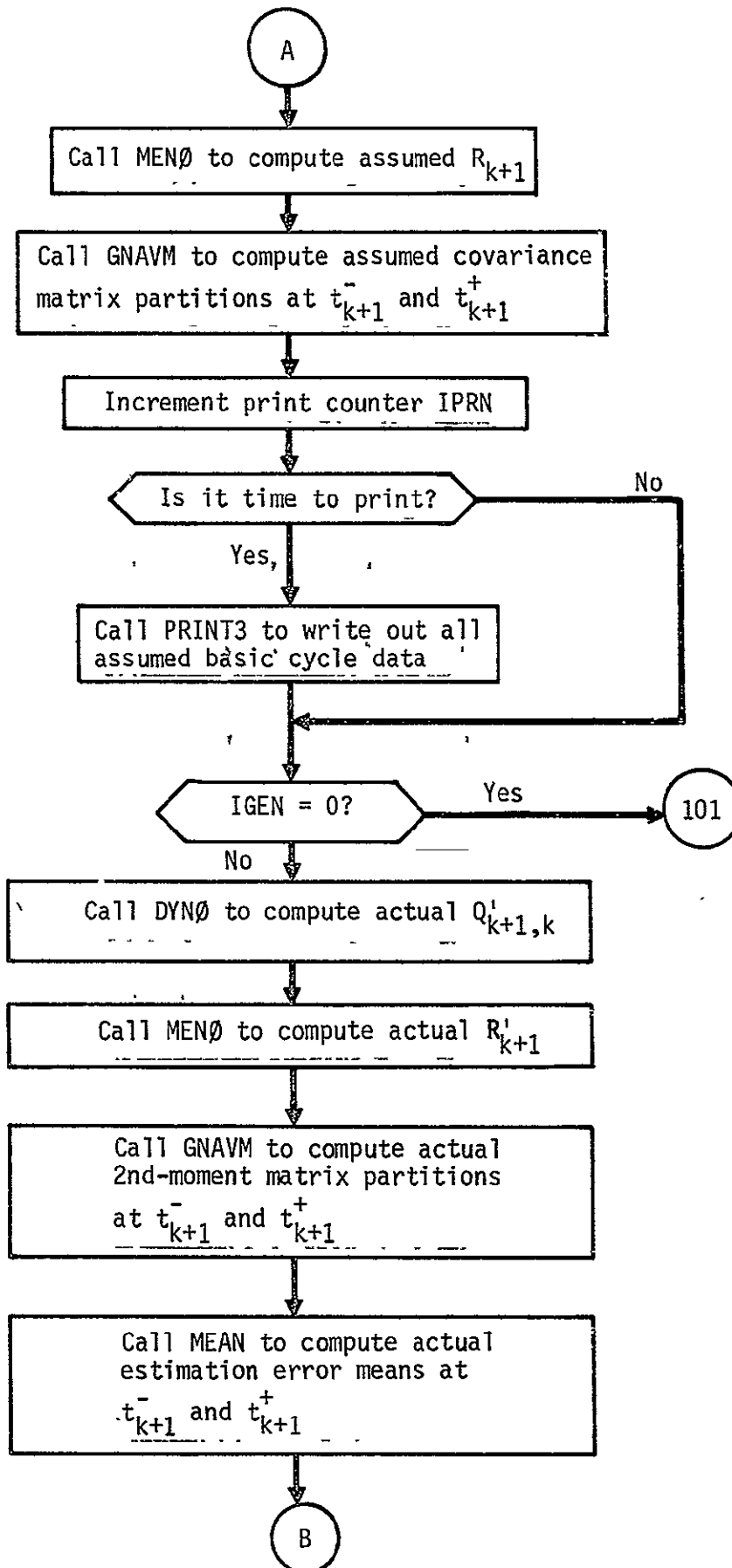
Subroutine ERRANN controls the computational flow through the basic cycle (measurement processing) and all events in the error analysis/generalized covariance analysis program.

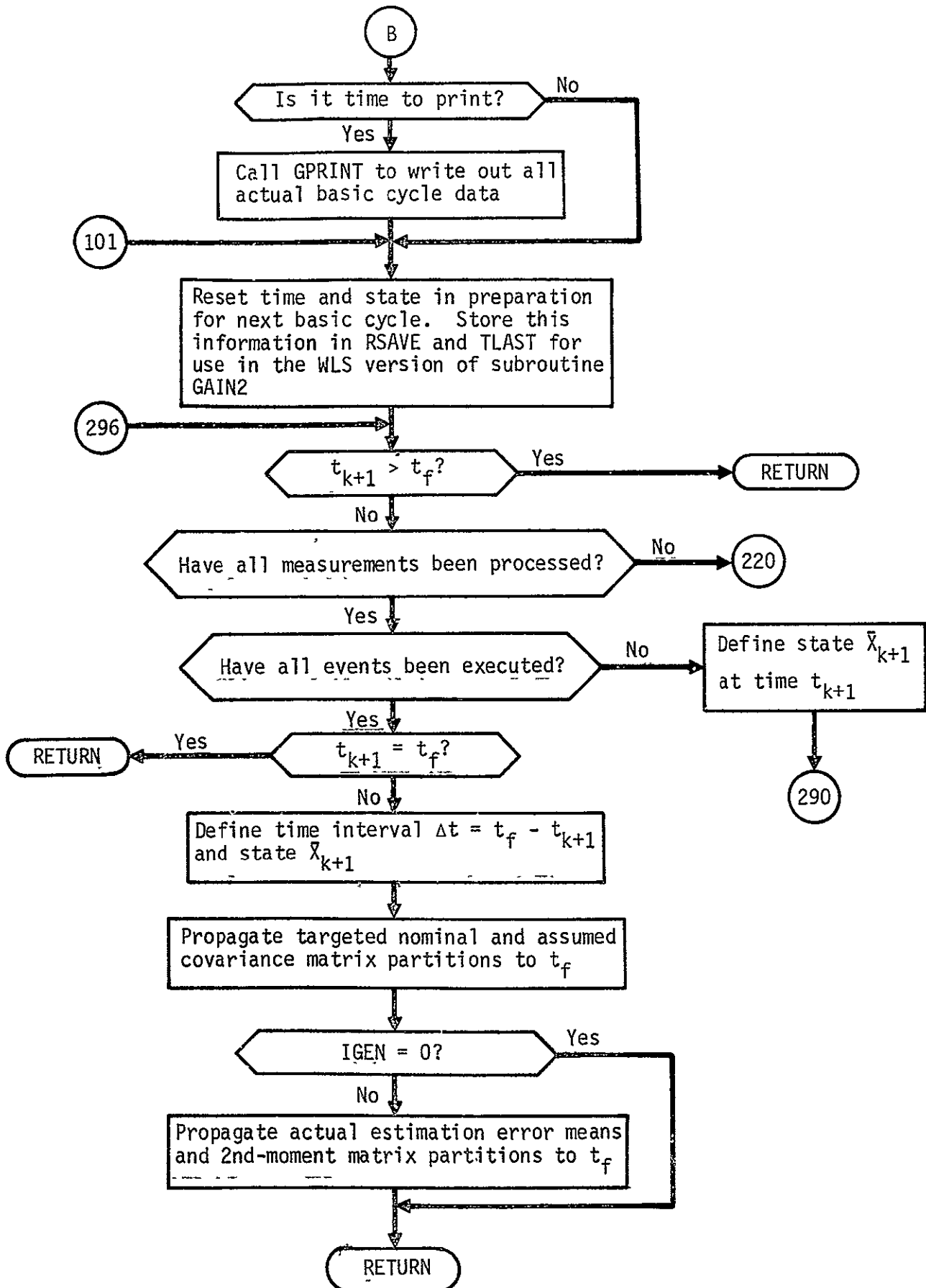
In the basic cycle the first task of ERRANN is to control the generation of the targeted nominal spacecraft state \bar{X}_{k+1} at time t_{k+1} , given the state \bar{X}_k at time t_k . Then calling PSIM, DYNØ, TRAKM, and MENØ, successively, ERRANN controls the computation of all matrix information required by subroutine GNAVM to compute the actual and assumed knowledge covariance matrix partitions at time t_{k+1}^+ immediately following the measurement.

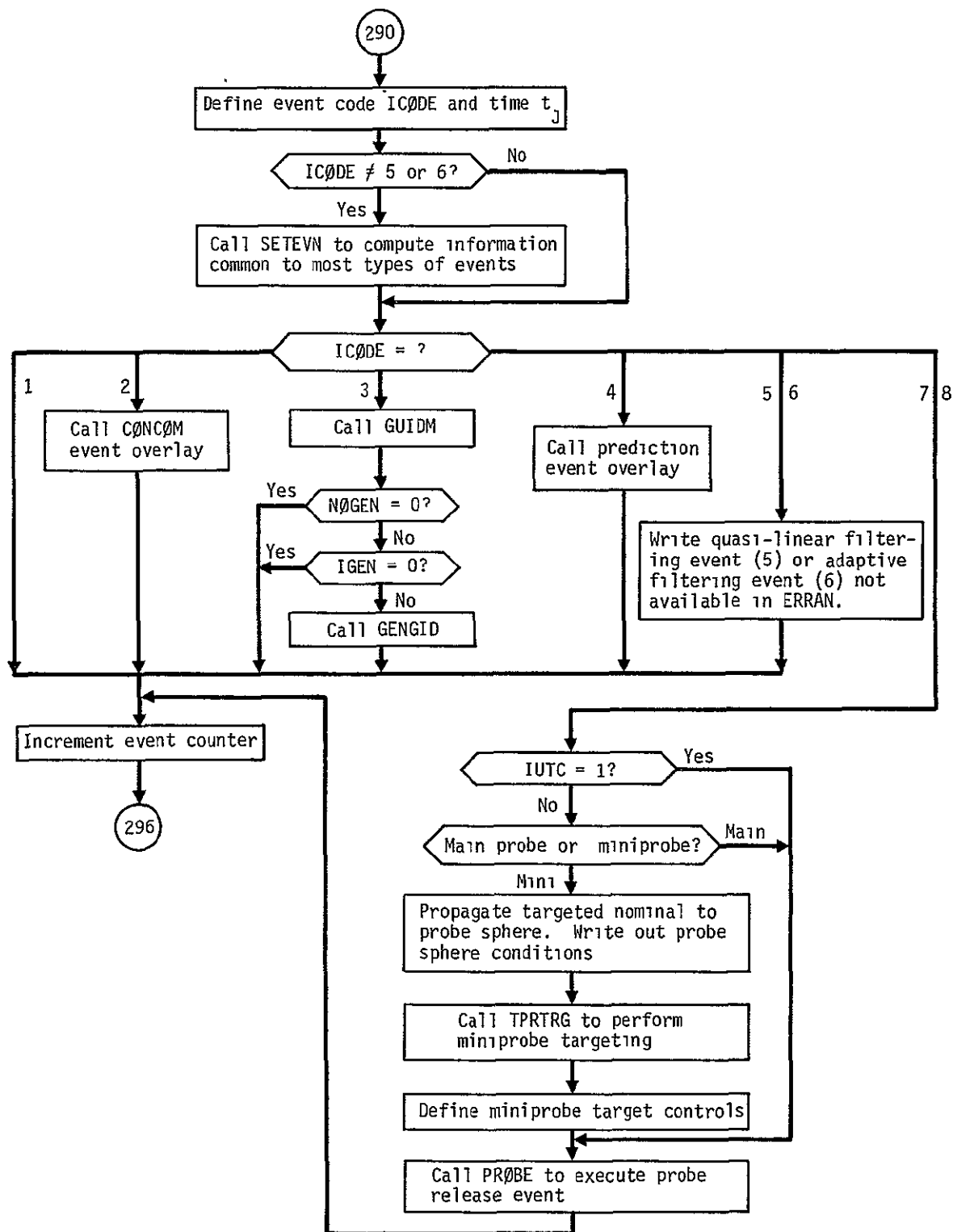
At an event, ERRANN simply calls the proper event subroutine or overlay where all required computations are performed. Subroutine ERRANN also controls miniprobe targeting in the error analysis program.

ERRANN Flow Chart









PROGRAM ERRON

PURPOSE: TO CONTROL THE ERROR ANALYSIS OVERLAY SCHEME

SUBROUTINES SUPPORTED: NONE

SUBROUTINES REQUIRED: DATA ERRAN PRNTS3

LOCAL SYMBOLS: IRUNX TOTAL NUMBER OF DATA CASES

IRUN DATA CASE COUNTER

SUBROUTINE ESTMT

PURPOSE: TO UPDATE THE FINAL VALUES OF THE PRECEDING COMPUTATION INTERVAL WHICH SERVE AS INITIAL VALUES FOR THE NEW STEP, TO DETERMINE THE DESIRED SIZE OF THE NEXT TIME INCREMENT ON THE BASIS OF TRUE ANOMALY OR REQUESTED PRINTTIME, AND TO ESTIMATE THE FINAL POSITION AND MAGNITUDE OF THE VIRTUAL MASS.

CALLING SEQUENCE: CALL ESTMT(D1,DELTM,TRTM)

ARGUMENTS	D1	I	JULIAN DATE, EPOCH 1900, OF THE INITIAL TRAJECTORY TIME
	DELTM	I	TIME INTERVAL OVER WHICH THE TRAJECTORY WILL BE PROPAGATED (DAYS)
	TRTM	I	INITIAL TRAJECTORY TIME (DAYS) REFERENCED TO INJECTION

SUBROUTINES SUPPORTED: VMP

SUBROUTINES REQUIRED: NONE

COMMON COMPUTED/USED: INCMNT V

COMMON COMPUTED: ITRAT KOUNT

COMMON USED: INCPR INC IPR

ESTMT Analysis

The initial values of the state variables are first set equal to the values at the end of the previous interval. The nominal time interval to be used during the current step is computed from

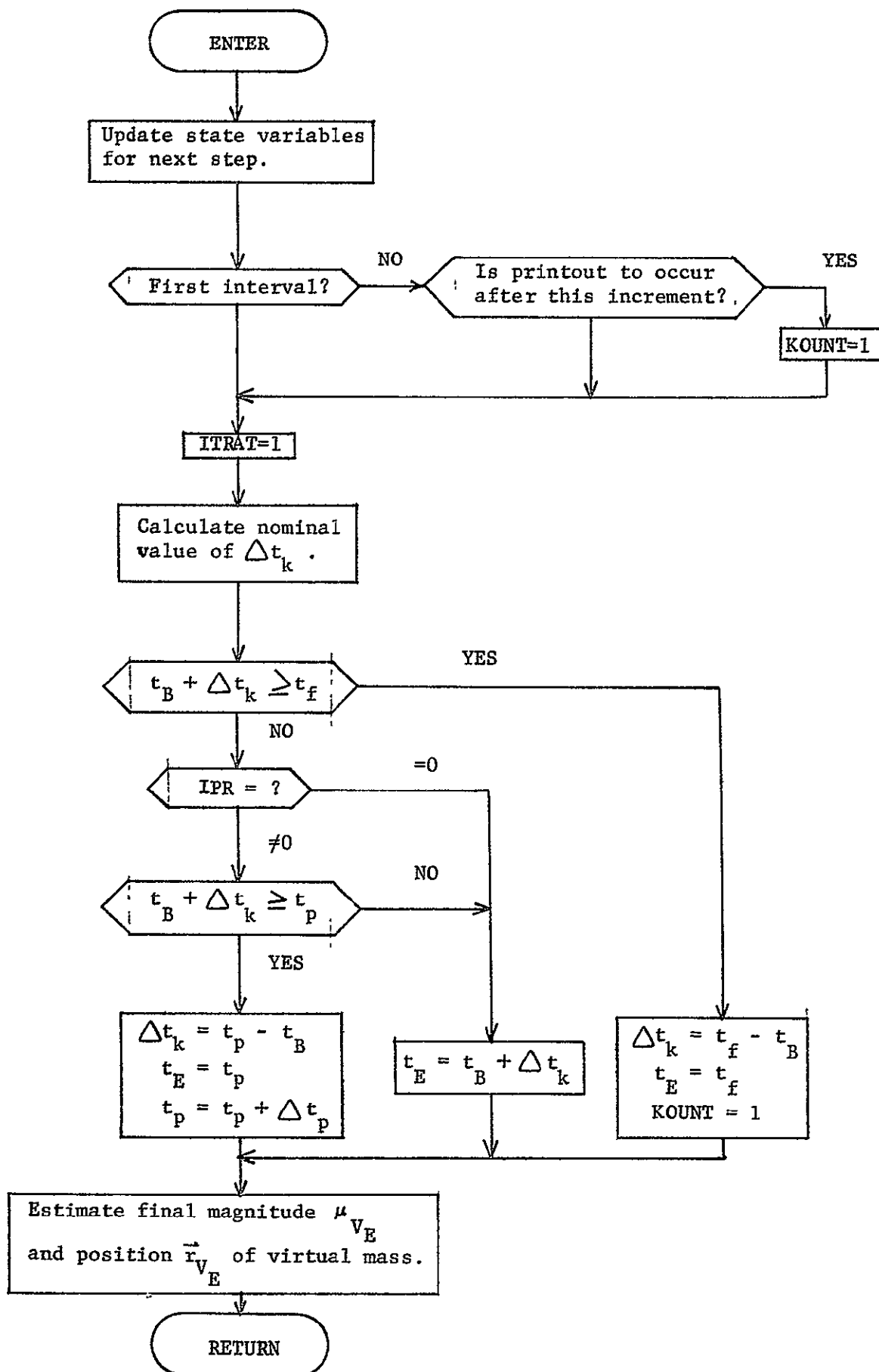
$$\Delta t_k = \frac{c_2 r_{VS_B}}{v_{VS_B}} \quad (1)$$

where c_2 is the constant input true anomaly increment relative to the virtual mass trajectory.

The time interval to the final time t_f or to the next time printout t_p is computed and the current time interval Δt is adjusted if necessary.

Finally the virtual mass final position and magnitude are estimated by the expansions

$$\begin{aligned} \mu_{V_E} &= \mu_{V_B} + \dot{\mu}_{V_B} \Delta t + \ddot{\mu}_V \Delta t^2 \\ \vec{r}_{V_E} &= \vec{r}_{V_B} + \dot{\vec{r}}_{V_B} \Delta t + \ddot{\vec{r}}_{V_{av}} \Delta t^2 \end{aligned} \quad (2)$$



SUBROUTINE EULMX

PURPOSE: TO COMPUTE THE MATRIX REQUIRED TO DEFINE TRANSFORMATIONS FROM ONE COORDINATE SYSTEM TO ANOTHER.

CALLING SEQUENCE: CALL EULMX(ALP,NN,BET,MM,GAM,LL,P)

ARGUMENT: ALP I FIRST ROTATION ANGLE (RADIAN)
 NN I FIRST AXIS OF ROTATION
 BET I SECOND ROTATION ANGLE (RADIAN)
 MM I SECOND AXIS OF ROTATION
 GAM I THIRD ROTATION ANGLE (RADIAN)
 LL I THIRD AXIS OF ROTATION
 P(3,3) O TRANSFORMATION MATRIX

SUBROUTINES SUPPORTED: PECEQ

SUBROUTINES REQUIRED: NONE

LOCAL SYMBOLS: A INTERMEDIATE ROTATION MATRIX
 ALPHA TEMPORARY LOCATION FOR EACH OF THE
 ROTATION ANGLES: ALP, BET, AND GAM
 D INTERMEDIATE PRODUCT MATRIX
 F TRANSFORMATION MATRIX FOR ANGLE ALP
 G TRANSFORMATION MATRIX FOR ANGLE BET
 H TRANSFORMATION MATRIX FOR ANGLE GAM
 N COUNTER SHOWING NUMBER OF COORDINATE AXES
 FOR WHICH CALCULATIONS REMAIN
 NAXIS TEMPORARY LOCATION FOR EACH OF THE AXES
 OF ROTATION: NN,MM, AND LL

COMMON USED: ONE ZERO

SUBROUTINE EXCUT

PURPOSE CONTROL EXECUTION OF A VELOCITY CORRECTION MODELED AS AN
IMPULSE SERIES IN THE ERROR ANALYSIS PROGRAM

CALLING SEQUENCE: CALL EXCUT

SUBROUTINES SUPPORTED: GUIDM

SUBROUTINES REQUIRED: PREPUL PULCOV PULSEX

COMMON COMPUTED/USED: XXIN

COMMON COMPUTED: QK

COMMON USED: DELPX, DIPX, TM, INPX

SUBROUTINE EXCUTE

PURPOSE: TO CONTROL THE ACTUAL EXECUTION OF THE VELOCITY INCREMENT DELTAV.

CALLING SEQUENCE: CALL EXCUTE

SUBROUTINES SUPPORTED: GIDANS

SUBROUTINES REQUIRED: PREPUL PULSEX CAREL RECEQ

LOCAL SYMBOLS:	A	SEMI-MAJOR AXIS OF DOMINANT BODY CONIC
	DVM	MAGNITUDE OF VELOCITY INCREMENT
	E	ECCENTRICITY OF DOMINANT BODY CONIC
	INDEX	CODE OF BODY BEING TESTED FOR DOMINANT BODY
	IND	INDEX OR CODE OF DOMINANT BODY
	ISUN	SUN VALUE OF IND
	I	INDEX
	JX	INDEX OF S/C-REL-TO-BODY ROW OF F-ARRAY
	MODEL	EXECUTION MODEL (1=IMPULSIVE, 2=PULSE ARC)
	PP	UNIT VECTOR TO PERIAPSIS IN ORBITAL PLANE
	QQ	UNIT VECTOR NORMAL TO PP IN ORBITAL PLANE
	RN	POSITION AND VELOCITY OF S/C AT END OF EXECUTION BY PULSING ARC
	RSI	POSITION VECTOR OF S/C RELATIVE TO DOMINANT BODY AT EXECUTION TIME
	RTB	RADIUS MAGNITUDE TO BODY BEING TESTED FOR DOMINANT BODY
	TA	TRUE ANOMALY ON DOMINANT BODY CONIC
	TFP	TIME FROM PERIAPSIS ON DOMINANT BODY CONIC
	VSI	VELOCITY VECTOR OF S/C RELATIVE TO DOMINANT BODY AT EXECUTION TIME
	WW	UNIT NORMAL TO ORBITAL PLANE
	W	ARGUMENT OF PERIAPSIS OF DOMINANT BODY

CONIC

XI INCLINATION OF DOMINANT BODY CONIC
 XMU GRAVITATIONAL CONSTANT OF DOMINANT BODY
 XN LONGITUDE OF ASCENDING NODE OF DOMINANT
 BODY

COMMON COMPUTED/USED: DELTAV D1 RIN TRTH

COMMON COMPUTED: KTIM

COMMON USED: ALNGTH DELV F KUR MDL
 NBOD NB PMASS PULT SPHERE
 TM TWO V

EXCUTE Analysis

EXCUTE is the executive subroutine controlling the actual execution of the velocity increment $\Delta \vec{v}$. The $\Delta \vec{v}$ is computed by TARGET or INSERS or read in by the user.

Before executing the correction EXCUTE computes peripheral information of interest to the user. It first determines the dominant body acting on the spacecraft. If the spacecraft is in the moon's SOI (with respect to the earth), the moon is the dominant body. If not in the moon's SOI but in any of the planets' SOI (with respect to the sun) that planet is the dominant body. Otherwise the sun is the dominant body.

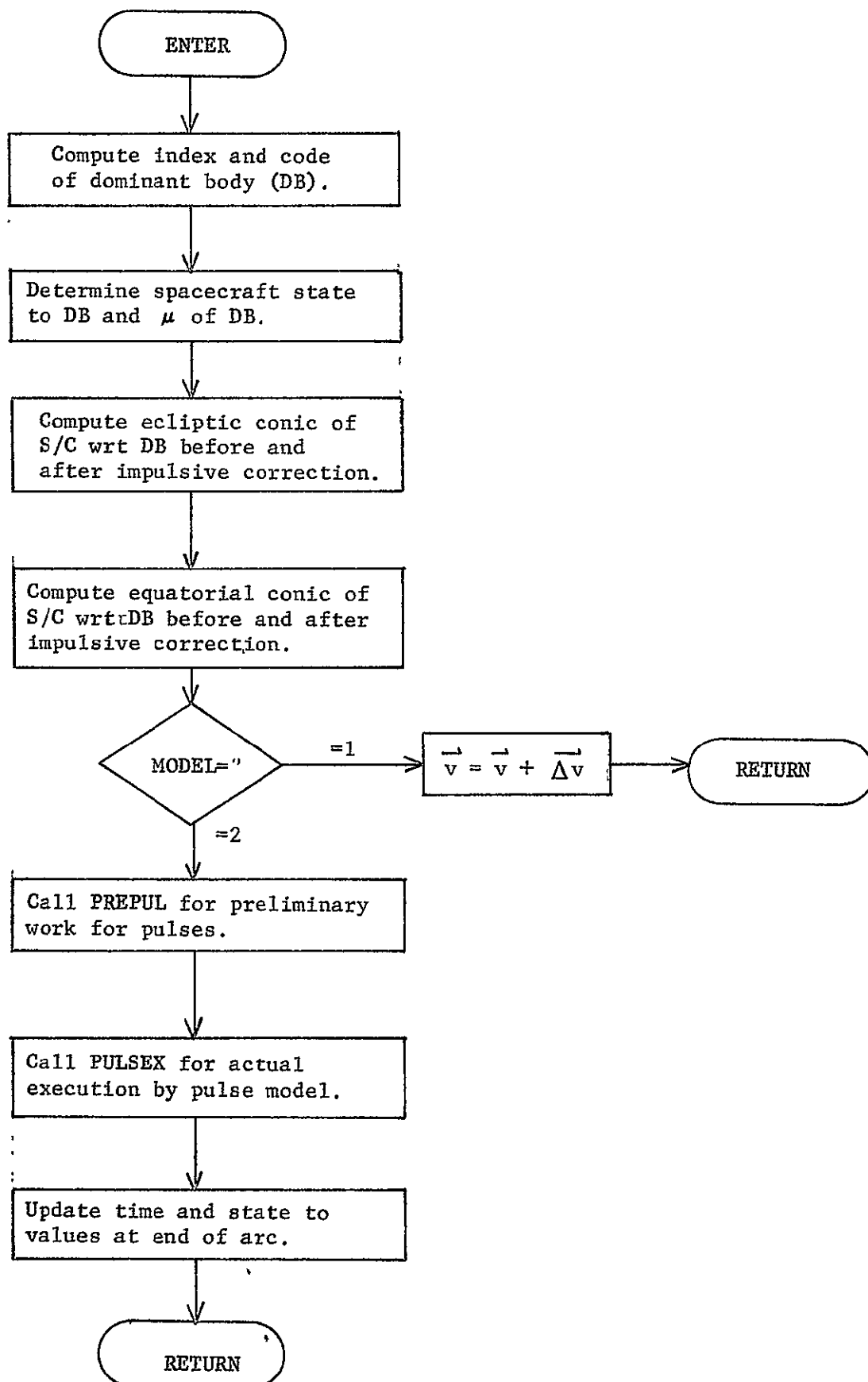
Having determined the dominant body EXCUTE computes the state of the spacecraft relative to that body. It then computes the conic elements of the trajectory both before and after an impulsive addition of the $\Delta \vec{v}$ in ecliptic coordinates.

If the dominant body is not the sun, it makes the same computations in equatorial coordinates.

EXCUTE then operates on the current value MODEL of the array MDL. If MODEL = 1, the impulsive model of execution is commanded. The $\Delta \vec{v}$ is therefore added to the current inertial ecliptic velocity before returning to GIDANS.

If MODEL = 2, the pulsing arc model of execution is required. PREPUL is called to perform the preliminary work needed for the pulsing arc. PULSEX then actually propagates the trajectory through the series of pulses. At the completion of the arc EXCUTE updates the time and inertial ecliptic state (both position and velocity) of the nominal trajectory to the state determined by PULSEX.

EXCUTE Flow Chart



SUBROUTINE EXCUTS

**PURPOSE CONTROL EXECUTION OF A VELOCITY CORRECTION MODELED AS AN
IMPULSE SERIES IN THE SIMULATION PROGRAM**

CALLING SEQUENCE: CALL EXCUTS

SUBROUTINES SUPPORTED: GUISIM

SUBROUTINES REQUIRED: PREPUL, PULCOV, PULSEX

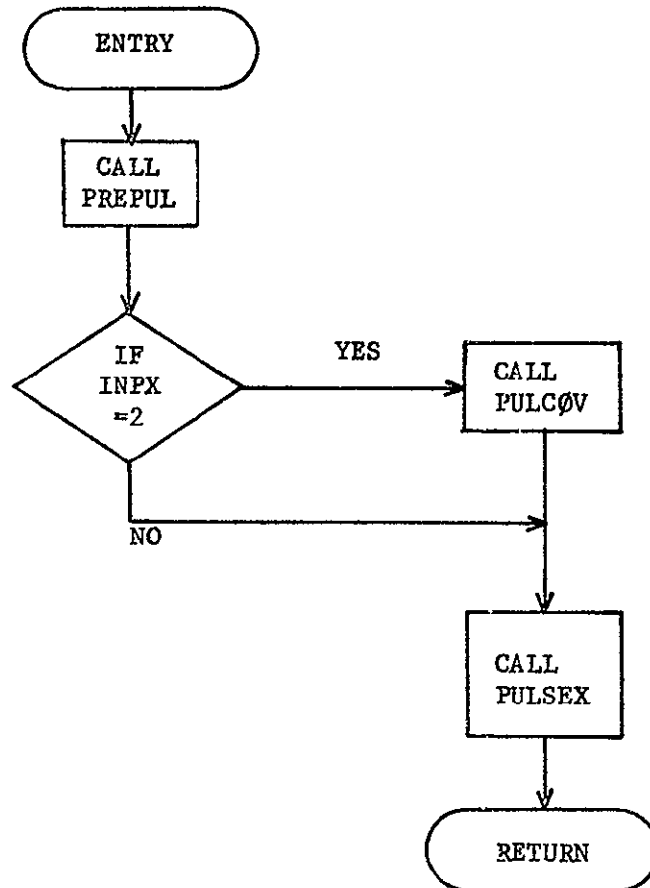
**LOCAL SYMBOLS RN EFFECTIVE SPACECRAFT STATE AFTER A
VELOCITY CORRECTION MODELED AS AN IMPULSE
SERIES**

COMMON COMPUTED/USED: XXIN

COMMON COMPUTED: QK

COMMON USED: DELPX, DIPX, TM, INPX

EXCUTS Flow Chart



FUNCTION F1

PURPOSE: TO CALCULATE SQUARE OF LENGTH OF CONSTRAINT VECTOR AFTER
TAKING STEP OF GIVEN LENGTH AND IN GIVEN DIRECTION
FROM CURRENT CONTROL VECTOR

ARGUMENT:	ALPHA	I	STEP LENGTH IN SEARCH DIRECTION
	FPHI	I	FUNCTION RELATING CONSTRAINT VECTOR TO CONTROL VECTOR
	M	I	DIMENSION OF CONSTRAINT VECTOR
	N	I	CONTROL VECTOR
	PHI	I	CURRENT CONSTRAINT VECTOR
	P	I	UNIT VECTOR IN SEARCH DIRECTION
	X	I	CURRENT CONTROL VECTOR

SUBROUTINES SUPPORTED: GAUSLS

SUBROUTINES REQUIRED: FPHI

LOCAL SYMBOLS:	F1	SQUARE OF LENGTH OF CONSTRAINT VECTOR AFTER TAKING PRESCRIBED STEP
	XALPHA	VECTOR OBTAINED FROM ORIGINAL CONTROL VECTOR BY TAKING STEP OF LENGTH ALPHA IN DIRECTION P

SUBROUTINE FLITE

PURPOSE: TO SOLVE THE TIME OF FLIGHT EQUATION (LAMBERT-S THEOREM)
USING BATTIN-S UNIVERSAL EQUATION FORMULATION.

CALLING SEQUENCE: CALL FLITE(R1,R2,THETA,GM,TF,A,E,K)

ARGUMENTS:

R1	I	INITIAL RADIUS
R2	I	FINAL RADIUS
THETA	I	CENTRAL ANGLE
GM	I	GRAVITATIONAL CONSTANT
TF	I	TIME OF FLIGHT
A	O	SEMI-MAJOR AXIS
E	O	ECCENTRICITY
K	O	ERROR CODE =0 NO ERROR =1 ERROR

SUBROUTINES SUPPORTED: HELIO

SUBROUTINES REQUIRED: NONE

LOCAL SYMBOLS:

AMISS	ERROR IN ITERATE
BIGNO	CONSTANT = $E+25$
B1	CONSTANT = $S^{3/2}$
CHECK	ERROR IN ITERATE
CX	BATTIN C-FUNCTION OF X
CY	BATTIN C-FUNCTION OF Y
C	CHORD LENGTH
DEM	INTERMEDIATE VARIABLE
P	SEMI-LATUS RECTUM
ROOT	INTERMEDIATE VARIABLE
SLOP	VALUE OF DERIATIVE OF T(X)
SX	BATTIN S-FUNCTION OF X

SY BATTIN S-FUNCTION OF Y
S1 SEMIPERIMETER
S =INTERMEDIATE VARIABLE ($=1-C/S1$)
TIME FLIGHT TIME CORRESPONDING TO ITERATE X
T NORMALIZED TIME OF FLIGHT
U FLAG SET TO 1 IF X LESS THAN $\pi/2$, -1 ELSE
VB1 INTERMEDIATE VARIABLE
V FLAG SET TO 1 FOR TYPE I, -1 FOR TYPE II
X1 , STARTING VALUE FOR X
X VARIABLE INTRODUCED TO REPLACE A
Y INTERMEDIATE VARIABLE AS FUNCTION OF X

FLITE Analysis

FLITE solves the time of flight equation (Lambert's theorem) using Battin's universal equation formulation. Stated functionally Lambert's theorem states that the time of flight t_f is a function

$$t_f = t_f(r_1 + r_2, c, a) \quad (1)$$

solely of the sum $r_1 + r_2$ of the distances of the initial and final points of the trajectory from the central body, the length c of the chord joining these points, and the length of the semimajor axis a of the trajectory. Usually the time of flight is known and it is desired to solve for the semimajor axis. The standard formulation involves different equations for the elliptic, parabolic, and hyperbolic cases, all of which then iterate on a to determine the solution.

In Battin's approach the semimajor axis a is replaced by a new variable x . By further introducing two new transcendental functions $S(x)$ and $C(x)$, the special cases of the flight-time equation are combined into one single, better behaved formula. The functions $S(x)$ and $C(x)$ are defined by

$$\begin{aligned} S(x) &= \frac{\sqrt{x} - \sin \sqrt{x}}{3} & C(x) &= \frac{1 - \cos \sqrt{x}}{x} & x > 0 \\ &= \frac{\sinh \sqrt{-x} - \sqrt{-x}}{\sqrt{x}^3} & &= \frac{\cosh \sqrt{-x} - 1}{-x} & x < 0 \\ &= \frac{1}{6} & &= \frac{1}{2} & x = 0 \end{aligned} \quad (2)$$

A parameter Q is introduced as

$$Q = \frac{s - c}{s}$$

where $c = (r_1^2 + r_2^2 - 2r_1 r_2 \cos \theta)^{1/2}$

$$s = \frac{1}{2} (r_1 + r_2 + c) \quad (3)$$

The universal flight-time formula is

$$T = \frac{S(x)}{C^{3/2}(x)} + Q^{3/2} \frac{S(y)}{C^{3/2}(y)}$$

$$yC(y) = Q \times C(x) \quad (4)$$

where $T = \sqrt{\frac{\mu}{s^3}} t_f$. The choice of the upper or lower sign is

made according to whether the transfer angle θ is less or greater than 180° respectively.

The development of equations (4) is too long and complex to be given here. It may be obtained from the first reference listed below. The following steps of that reference are noted:

- (1) the two body problem on pp. 15,16
- (2) the "vis viva" equation and Kepler's equation on pp. 50,51
- (3) Lambert's theorem proved from Kepler's equation on p. 71
- (4) the basic flight-time formula and detailed analysis on pp. 72-78
- (5) The universal formulation on pp. 80,81.

Instead of using the equations (4) the authors of reference 2 (listed below) determined y as a function of x as

$$y = 4 \arcsin^2 \sqrt{\left| \frac{xsC(x)}{2} \right|} \quad x \geq 0$$

$$= -4 \ln^2 \left\{ \sqrt{\left| \frac{xsC(x)}{2} \right|} + \sqrt{\left| \frac{xsC(x)}{2} \right|} + 1 \right\}^{\frac{1}{2}} \quad x < 0 \quad (5)$$

Therefore a single variable iteration is possible. Newton's method is used to solve (4a) given T and Q as

$$x = x - \frac{T(x_n) - T}{T'(x_n)} \quad (6)$$

$$\text{where } T(x) = \frac{S(x)}{C^{3/2}(x)} + Q^{3/2} \frac{S(y)}{C^{3/2}(y)} \quad (7)$$

$$T'(x) = \frac{1 + k \left[+ Q^{3/2} - 1.5 \sqrt{2-yC(y)} T(x) \right]}{2x \sqrt{C(x)}} \quad (8)$$

$$k = \operatorname{sgn}(\pi^2 - x) \sqrt{\frac{2 - xC(x)}{2 - yC(y)}} \quad (9)$$

As $|2 - yC(y)| \rightarrow 0$, $k \rightarrow 1$. Therefore if $|2 - yC(y)| < 10^{-4}$ k is set to 1. Also $T'(x)$ breaks down as $x \rightarrow 0$. Therefore the approximation is used:

$$T'(x) = \frac{1 \mp Q^{3/2}}{2\pi^2} \quad |x| < 10^{-6} \quad (10)$$

The starting value for x is given by $x = x_1 - \Delta x(T, Q)$ where

$$\begin{aligned} x_1 &= 82.1678 + 352.8045 T \\ &\quad - (123954.8504 T^2 + 43904.0083 T + 13423.6819)^{1/2} \\ \Delta x(T, Q) &= \mp \left(\frac{2.36}{T^2 + .15} + \frac{3.4}{T + .1} \right) (0.3 Q^2 + 0.7 Q) \end{aligned} \quad (11)$$

To insure that the routine will not fail for large or small values of T certain restrictions on T are built into the program. The nominal value of T is forced to be no larger than 950,000 and no smaller than 10^{-6} . This forces the corresponding limits for x of $-823.0473 \leq x \leq 39.14553$.

Finally convergence is achieved when $|T(x_n) - T| < \frac{T}{100000}$.

Having solved for semimajor axis a , the semilatus rectum p is given by

$$p = \frac{1}{2} \left\{ \frac{r_1 r_2 \sin \theta}{c} \sqrt{\frac{1}{s-c} - \frac{1}{2a}} \pm \operatorname{sgn}(t_m - t) \sqrt{\frac{1}{s} - \frac{1}{2a}} \right\}^2 \quad (12)$$

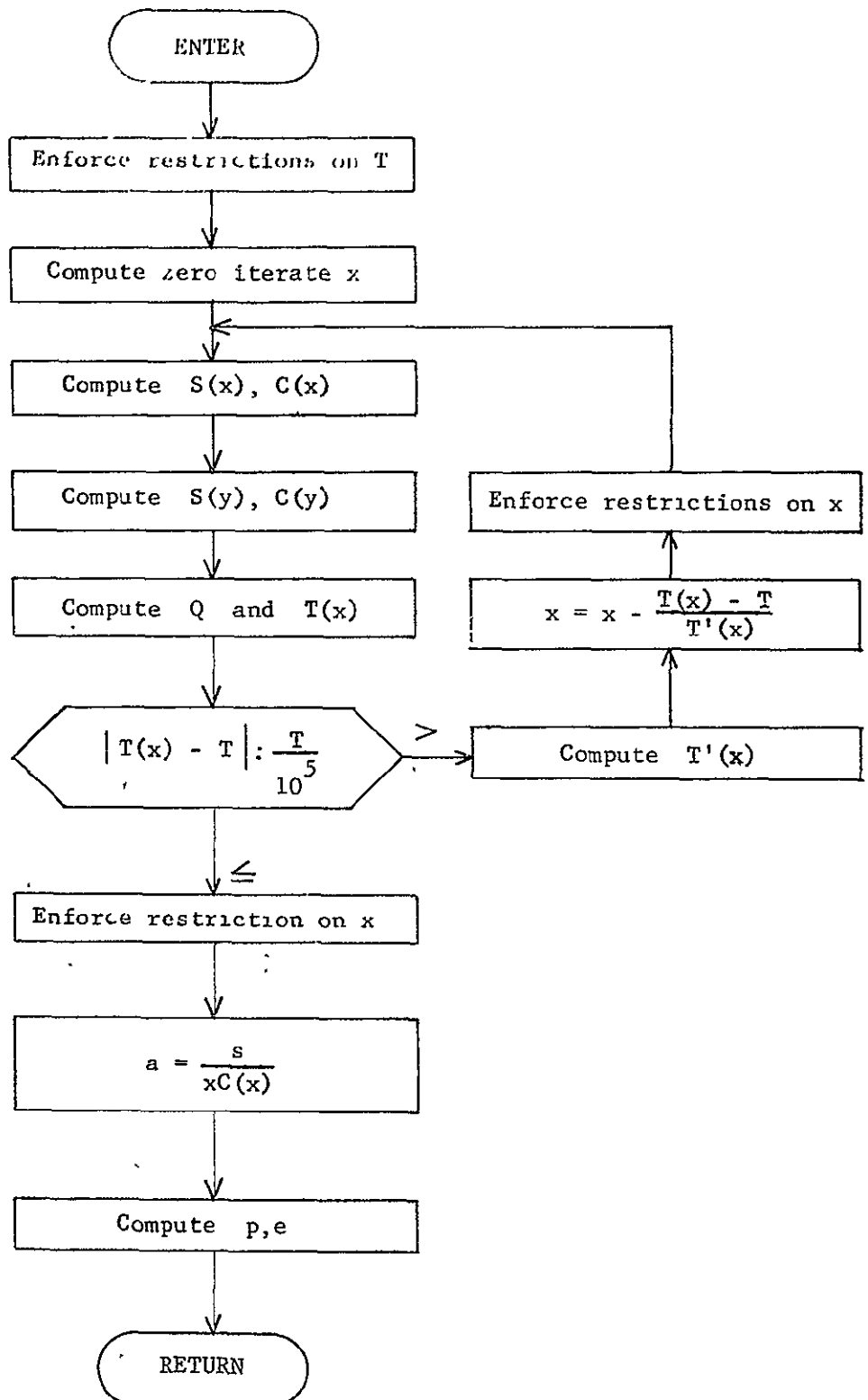
Then the eccentricity e is given by

$$e = 1 - \frac{p}{a} \quad (13)$$

References:

- (1) Battin, R. H., Astronomical Guidance, McGraw Hill Book Co., Inc., New York, 1964.
- (2) Lesh, H. F., and Travis, C., FLIGHT: a Subroutine to Solve the Flight Time Problem, JPL Space Programs Summary 37-53, Vol. II.

FLITE Flow Chart



SUBROUTINE GAIN1

PURPOSE: TO COMPUTE THE KALMAN GAIN MATRICES

CALLING SEQUENCE: CALL GAIN1(NR,AJ,AKW,SW,IEND)

ARGUMENTS: NR I NUMBER OF ROWS IN THE OBSERVATION MATRIX
AJ I MEASUREMENT RESIDUAL COVARIANCE AND ITS
INVERSE
AKW I INTERMEDIATE ARRAY
SW I INTERMEDIATE ARRAY
IEND I NR-1

SUBROUTINES SUPPORTED: GNAVM

SUBROUTINES REQUIRED: MATIN

LOCAL SYMBOLS: DUM INTERMEDIATE VECTOR
XJ INTERMEDIATE ARRAY
SUM INTERMEDIATE VARIABLE

COMMON COMPUTED: AK S

COMMON USED: ONE HALF ZERO

GAIN1 Analysis

Subroutine GAIN1 computes the Kalman-Schmidt filter gain matrices K_{k+1} and S_{k+1} that are used in subroutines GNAVM and NAVM to update estimation error covariance matrices after a measurement has been processed.

The measurement residual covariance matrix J_{k+1} and the auxiliary matrices A_{k+1} and B_{k+1} are assumed to be available (from GNAVM or NAVM) when GAIN1 is called. Subroutine GAIN1 then evaluates the following equations to determine the filter gain matrices:

$$K_{k+1} = A_{k+1} J_{k+1}^{-1} \quad (1)$$

$$S_{k+1} = B_{k+1} J_{k+1}^{-1} \quad (2)$$

SUBROUTINE GAIN2

PURPOSE: TO COMPUTE THE GAIN MATRICES FOR THE EQUIVALENT RECURSIVE
WEIGHTED LEAST SQUARES CONSIDER FILTER

CALLING SEQUENCE: CALL GAIN2(NR)

ARGUMENT: NR I NUMBER OF ROWS IN OBSERVATION MATRIX

SUBROUTINES SUPPORTED: NAVM GNAVM

SUBROUTINES REQUIRED: PSIM DYN0 MATIN

LOCAL SYMBOLS: AJ MEASUREMENT RESIDUAL COVARIANCE MATRIX AND
ITS INVERSE

AKW INTERMEDIATE VARIABLE

DELTMS INTERMEDIATE STORAGE FOR DELTM

DUM INTERMEDIATE VECTOR

IEND NR-1

IFLAG =0 IF STATE TRANSITION MATRICES USED ARE
IDENTICAL TO STM FOR THE MEASUREMENT

=1 IF STM ARE RECALCULATED TO CORRESPOND TO
THE DT BETWEEN THE LAST MEASUREMENT AND
THE PRESENT MEASUREMENT

NDIM2S TEMPORARY STORAGE FOR NDIM2

NDIM3S TEMPORARY STORAGE FOR NDIM3

PHISV TEMPORARY STORAGE FOR PHI

PSAVE INTERMEDIATE ARRAY

SUM INTERMEDIATE VARIABLE

SW INTERMEDIATE ARRAY

TRTS INTERMEDIATE STORAGE FOR TRTM1

TXUSV INTERMEDIATE STORAGE FOR TXU ARRAY

TXXSSV INTERMEDIATE STORAGE FOR TXXS ARRAY

COMMON COMPUTED/USED:	AK	CMIN	CPLU	DELTM	PHI
	PMIN	PPLU	PSMIN	PSPLU	TXXS

COMMON USED:	AM	CXXSG	H	II	MCNTR
	NDIM1	NDIM2	NDIM3	ONE	PG

GAIN2-B

PSG	RF	RSAVE	S	TE VN
TLAST	TRTM1	TXU	ZERO	

GAIN2 Analysis

Subroutine GAIN2 computes filter gain matrices K_{k+1} and S_{k+1} for an equivalent recursive weighted-least-squares (WLS) consider filter. The equations required to compute K_{k+1} and S_{k+1} are identical to those used to compute the Kalman filter gains, but with all consider parameter covariances removed.

Subroutine GAIN2 propagates and updates (at a measurement) a set of covariance matrix partitions that are completely independent of those processed in subroutines NAVM or GNAVM for the sole purpose of generating filter gain matrices K_{k+1} and S_{k+1} .

The propagation and update equations employed in GAIN2, which are a subset of the NAVM and GNAVM propagation and update equations, are summarized below. For definitions of all matrices, see either the subroutine NAVM or GNAVM analysis section.

Propagation equations:

$$P_{k+1}^- = \left(\Phi P_k^+ + \theta_{xx} C_{xx}^{+T} \right) \Phi^+ + C_{xx}^- \theta_{xx}^T + Q_k \quad (1)$$

$$C_{xx}^- = \Phi C_{xx}^+ + \theta_{xx} P_{s_k}^+ \quad (2)$$

$$P_{s_{k+1}}^- = P_{s_k}^+ \quad (3)$$

Gain equations:

$$A_{k+1} = P_{k+1}^- H_{k+1}^T + C_{xx}^- M_{k+1}^T \quad (4)$$

$$B_{k+1} = P_{s_{k+1}}^- M_{k+1}^T + C_{xx}^- H_{k+1}^T \quad (5)$$

$$J_{k+1} = H_{k+1} A_{k+1} + M_{k+1} B_{k+1} + R_{k+1} \quad (6)$$

$$K_{k+1} = A_{k+1} J_{k+1}^{-1} \quad (7)$$

$$S_{k+1} = B_{k+1} J_{k+1}^{-1} \quad (8)$$

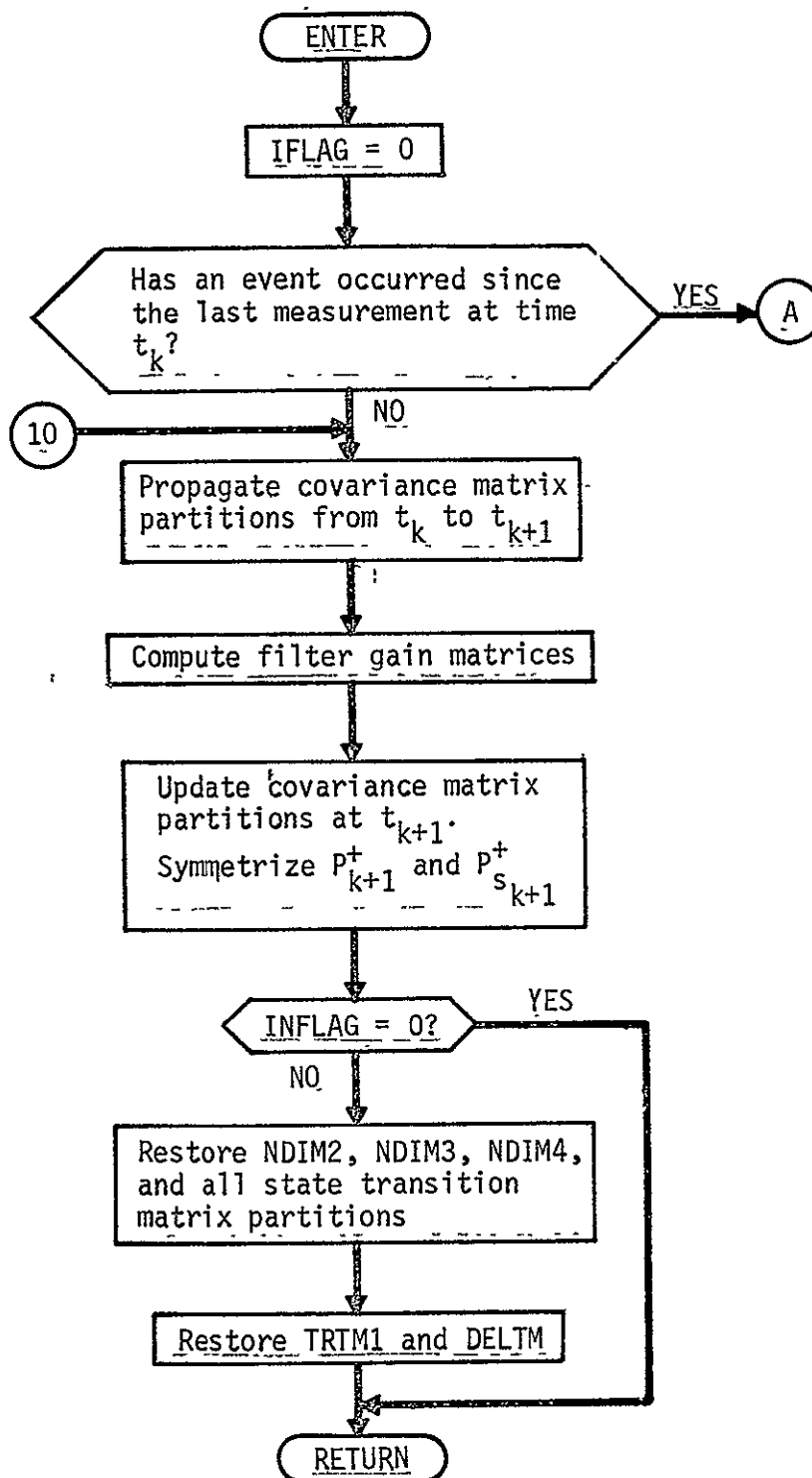
Update equations:

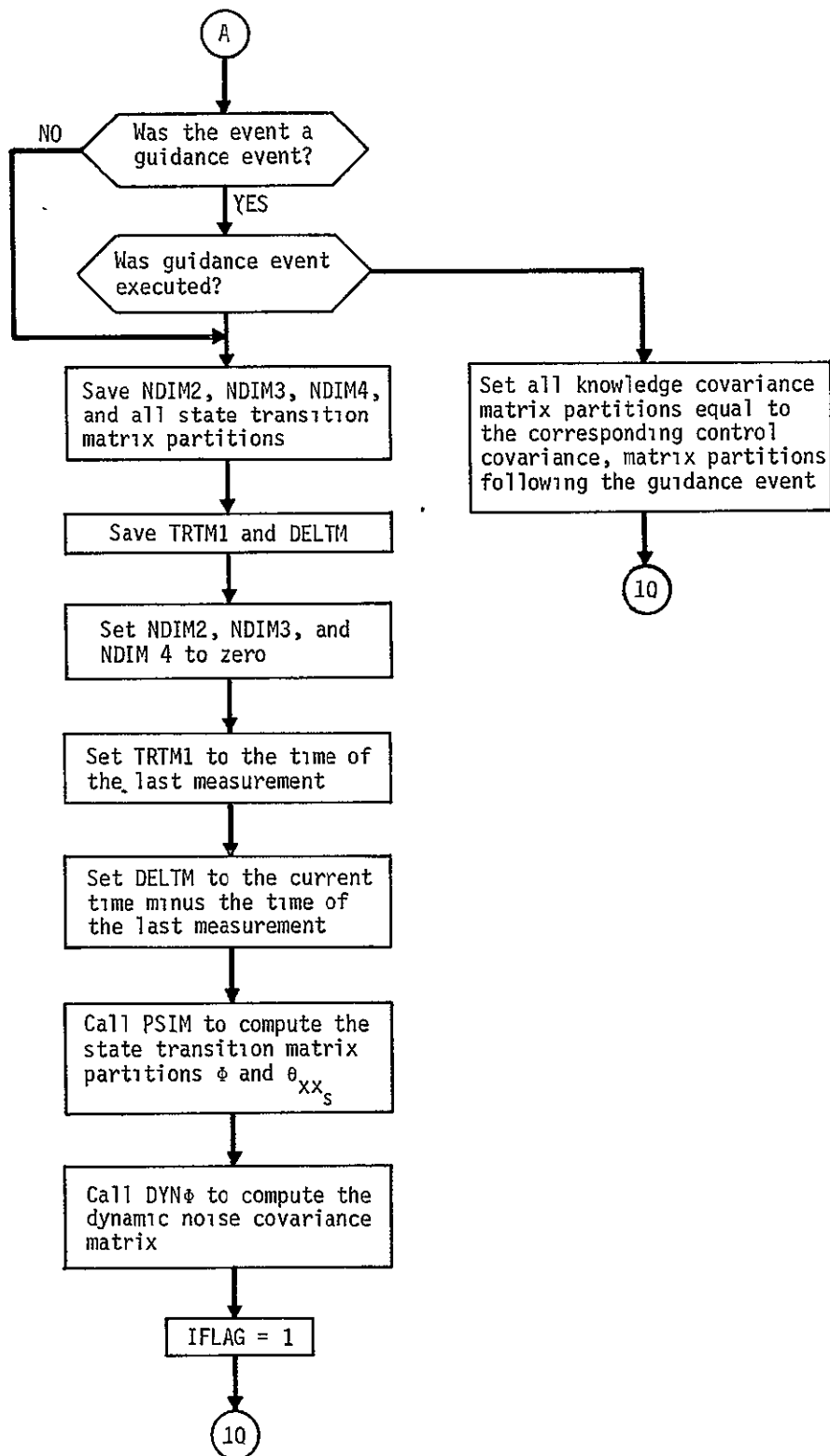
$$P_{k+1}^+ = P_{k+1}^- - K_{k+1} A_{k+1}^T \quad (9)$$

$$C_{xx, s_{k+1}}^+ = C_{xx, s_{k+1}}^- - K_{k+1} B_{k+1}^T \quad (10)$$

$$P_{s_{k+1}}^+ = P_{s_{k+1}}^- - S_{k+1} B_{k+1}^T \quad (11)$$

GAIN2 Flow Chart





SUBROUTINE GAUSLS

PURPOSE: TO FIND CONTROL VECTOR OF DIMENSION N WHICH MINIMIZES
SQUARE OF LENGTH OF CONSTRAINT VECTOR OF DIMENSION M

ARGUMENT: C1 I WEIGHTING FACTOR, APPLIED TO CHANGE IN
LENGTH OF CONTROL VECTOR IN CONVERGENCE
CRITERION

C2 I WEIGHTING FACTOR APPLIED TO CHANGE IN
MAGNITUDE OF MISS INDEX IN CONVERGENCE
CRITERION

DELTA I COMMON PERTURBATION SIZE APPLIED TO ALL
CONTROL COMPONENTS IN APPROXIMATING
JACOBIAN SENSITIVITY MATRIX BY DIVIDED
DIFFERENCES

EPS I UPPER BOUND ON WEIGHTED SUM OF CHANGES IN
CONTROL VECTOR AND IN MISS INDEX FOR
CONVERGENCE

FPHI I FUNCTION RELATING CONSTRAINT VECTOR TO
CONTROL VECTOR

ICONV1 0 FLAG INDICATING WHETHER OR NOT
CONVERGENCE OCCURRED
=1 CONVERGENCE
=2 NO CONVERGENCE

ITLIM I UPPER BOUND ON NUMBER OF PERMISSIBLE
ITERATIONS BEFORE ALGORITHM IS STOPPED

M I NUMBER OF CONSTRAINTS(DIMENSION OF
CONSTRAINT VECTOR)

N I NUMBER OF CONTROLS(DIMENSION OF CONTROL
VECTOR)

PHI I CONSTRAINT VECTOR

S0 I MAXIMUM PERMISSIBLE LENGTH OF PSEUDO-
INVERSE CONTROL STEP(LARGER STEPS ARE
REPLACED BY STEEPEST DESCENT CORRECTION)

X I/O CONTROL VECTOR

YM 0 FINAL MINIMUM VALUE OF MISS INDEX

SUBROUTINES SUPPORTED: TPRTRG

SUBROUTINES REQUIRED: FPHI JACOB MATIN THPOSM

LOCAL SYMBOLS:

ALPAM	PRODUCT OF ALPHA AND AMBOA
ALPHA	FACTOR DETERMINING LOCATION OF INTERMEDIATE FUNCTION EVALUATION ON INTERVAL FROM 0. TO AMBOA FOR CUBIC INTERPOLATION
AMBOA	LENGTH OF INTERVAL USED IN CUBIC INTERPOLATION TO FIND MINIMUM IN SEARCH DIRECTION
DXMS	MAGNITUDE SQUARED OF CONTROL CORRECTION
DXM	MAGNITUDE OF CONTROL CORRECTION
GRADYM	MAGNITUDE OF GRADIENT OF MISS INDEX
GRADY	GRADIENT OF MISS INDEX
HALF	CONSTANT 0.5
ILOC	INDEX USED IN TRANSFORMING A SQUARE MATRIX INTO A COLUMN VECTOR
ILS	FLAG INDICATING WHETHER PREVIOUS STEP WAS PSEUDO-INVERSE OR STEEPEST DESCENT CONTROL CORRECTION =1 PSEUDO-INVERSE =2 STEEPEST DESCENT
ITM1	CURRENT ITERATION NUMBER LESS ONE
ITP1	CURRENT ITERATION NUMBER PLUS ONE
IT	CURRENT ITERATION NUMBER
IYC	FLAG INDICATING WHETHER OR NOT MISS INDEX WAS CALCULATED FROM LAST CONTROL CORRECTION =1 MISS INDEX CALCULATED =2 MISS INDEX NOT CALCULATED
PROJM	PSEUDO-INVERSE MATRIX
PSDIJ	INVERSE OF PRODUCT OF JACOBIAN MATRIX BY JACOBIAN MATRIX ITSELF
PSDJ	PRODUCT OF TRANSPOSE OF JACOBIAN MATRIX BY JACOBIAN MATRIX ITSELF
RECORD	ARRAY CONTAINING CONTROL VECTORS, MISS INDICES, AND GRADIENTS OF MISS INDICES FOR ALL ITERATIONS

RJCBM	JACOBIAN SENSITIVITY MATRIX OF CONSTRAINT VECTOR WITH RESPECT TO CONTROL VECTOR
TEMPV	COLUMN VECTOR REPRESENTATION OF SQUARE MATRIX PSDJ
TEMPW	COLUMN VECTOR REPRESENTATION OF SQUARE MATRIX PSDIJ
TWO	CONSTANT 2.0
WINCS	WEIGHTED SUM OF CHANGE IN LENGTH OF CONTROL VECTOR AND MAGNITUDE MISS INDEX USED IN CONVERGENCE CRITERION
XMIN	LENGTH OF STEP WHICH MINIMIZES MISS INDEX IN SEARCH DIRECTION
YALAM	VALUE OF MISS INDEX AFTER STEP OF LENGTH ALPHA TIMES LAMBDA IN SEARCH DIRECTION
YLAM	VALUE OF MISS INDEX AFTER STEP OF LAMBDA IN SEARCH DIRECTION
YP	VALUE OF MISS INDEX ON PREVIOUS ITERATE
Y	CURRENT VALUE OF MISS INDEX
ZERO	CONSTANT 0.

GAUSLS Analysis

GAUSLS is a hybrid algorithm to obtain a least-squares solution to the system

$$\underline{\phi}(\underline{x}) = 0 \quad (1)$$

where \underline{x} is an n -dimensional control vector, $\underline{\phi}$ is an m -dimensional constraint vector, and $m \geq n$. Current array dimensions in GAUSLS require that $n \leq 5$ and $m \leq 10$. By least-squares we mean that the square of the standard Euclidean norm of $\underline{\phi}$, namely

$$||\underline{\phi}||^2 = \underline{\phi}^T \underline{\phi} \quad , \quad (2)$$

is minimized. The principal algorithm used is the well-known pseudo-inverse scheme originally due to Gauss. When, however, the dependence of $\underline{\phi}$ on \underline{x} deviates substantially from the approximate linearity tacitly assumed by the Gauss method, a best-step steepest descent algorithm is invoked. Either of two indications of nonlinearity can cause GAUSLS to transfer from the normal pseudo-inverse mode to best-step steepest descent technique: (1) the Gauss control correction is larger in norm than an input upper bound, s_0 , or (2) the Gauss step actually increases the miss index, $||\underline{\phi}||^2$, over the previous iterate.

The Gauss procedure can readily be derived since it is simply the exact one-step solution to equation (1) when $\underline{\phi}$ depends on \underline{x} linearly. Let J represent the Jacobian or sensitivity matrix of $\underline{\phi}$ with respect to \underline{x} ; that is

$$J_{ij} = \frac{\partial \phi_i}{\partial x_j} \quad \begin{matrix} i=1, \dots, m \\ j=1, \dots, n \\ m \geq n \end{matrix} \quad . \quad (3)$$

Next let y denote the least-squares miss index; that is

$$y = ||\underline{\phi}||^2 \quad (4)$$

Then the gradient of the miss-index is simply

$$\nabla y = 2 J^T \underline{\phi} \quad (5)$$

Now a necessary condition for the miss-index to be minimized after a control correction of $\Delta \underline{x}$ is

$$\nabla y (\underline{x} + \Delta \underline{x}) = 0 \quad (6)$$

Substituting equation (5) into (6) gives

$$2 J^T (\underline{x} + \Delta \underline{x})^T \underline{\phi} (\underline{x} + \Delta \underline{x}) = 0 \quad (7)$$

Assuming J either constant or approximately so and using the first two terms of the Taylor's series for $\underline{\phi}$ yields the approximation

$$2 J^T [\underline{\phi}(\underline{x}) + J \Delta \underline{x}] = 0 \quad (8)$$

Solving for the control correction then yields the pseudo-inverse control correction

$$\Delta \underline{x} = -(J^T J)^{-1} J^T \underline{\phi}(\underline{x}) \quad (9)$$

Clearly equation (9) is exact if $\underline{\phi}$ is a linear function of \underline{x} so that the Taylor series of $\underline{\phi}(\underline{x} + \Delta \underline{x})$ has only two terms and J is independent of \underline{x} . Since one can reasonably expect that if the dependence of $\underline{\phi}$ on \underline{x} is approximately linear, formula (9) can be applied iteratively to yield a convergent sequence of control vectors converging to the least-squares solution and one arrives at the Gauss algorithm, namely,

$$\Delta \underline{x}_k = (J^T J)^{-1} J^T \underline{\phi}(\underline{x}_k) \quad (10)$$

$$k = 0, 1, 2, \dots$$

$$\underline{x}_{k+1} = \underline{x}_k + \Delta \underline{x}_k \quad (11)$$

where \underline{x}_0 is an initial control estimate suggested by other sources.

GAUSLS requires as an input parameter this zero-iterate control estimate, together with the corresponding constraints $\phi(\underline{x}_0)$.

Since equation (6) only guarantees an extremum of the miss index (i.e., a minimum, a maximum, or an inflection point), no more can be said for the Gauss algorithm. It must be assumed that the initial estimate of \underline{x}_0 is sufficiently near a local minimum and that y is well enough behaved that the algorithm indeed leads to that minimum. It is interesting to note that in the case that $m = n$, equations (10) and (11) reduce to the familiar Newton-Raphson scheme for solving nonlinear systems of equations.

The logic behind the steepest-descent mode is less elegant but more straightforward than the Gauss procedure. First, the gradient of the miss index is computed via equation (5). Next a search is conducted in the negative gradient direction until the miss index is observed to begin increasing. Let λ denote the step length in the search direction where y is first observed to increase. Then the subroutine THPØSM is called to find a minimum of y on the step length interval from 0 to λ by cubic interpolation. Let λ_m denote the step length value corresponding to the minimum returned by THPØSM. Then the control correction for the k th iterate is taken to be

$$\Delta \underline{x}_k = -\lambda_m \underline{\nabla} y / \|\underline{\nabla} y\| \quad (12)$$

The convergence of this scheme is only asymptotic with no acceleration as the minimum-miss controls are approached. Nevertheless, the steepest descent algorithm seems to be the best available for extremely nonlinear miss indices since it involves no linear extrapolation and since it searches in the only direction in which improvement is guaranteed. Its poor terminal convergence is no handicap in the hybrid GAUSLS routine because once the iteration sequence falls inside a suitably linear region about the miss-index minimum, the rapidly convergent Gauss scheme takes over.

GAUSLS calculates the Jacobian matrix J by numerical differencing through a call to the subroutine JACØB. Hence the user is required to supply a perturbation size δ to GAUSLS for use by JACØB in approximating J by the forward-divided difference

$$J_{ij} \approx \frac{[\phi_i(\underline{x}_j + \delta) - \phi_i(\underline{x}_j)]}{\delta} \quad (13)$$

The user could conceivably use an analytical Jacobian matrix by replacing the call to JACØB by formulae for the appropriate partial derivatives.

The convergence criterion in either mode of GAUSLS is the same. Adequate convergence is assumed when a weighted sum of the length of the change in the control vector and the magnitude of the change in the miss index fall below a preassigned value; i.e.,

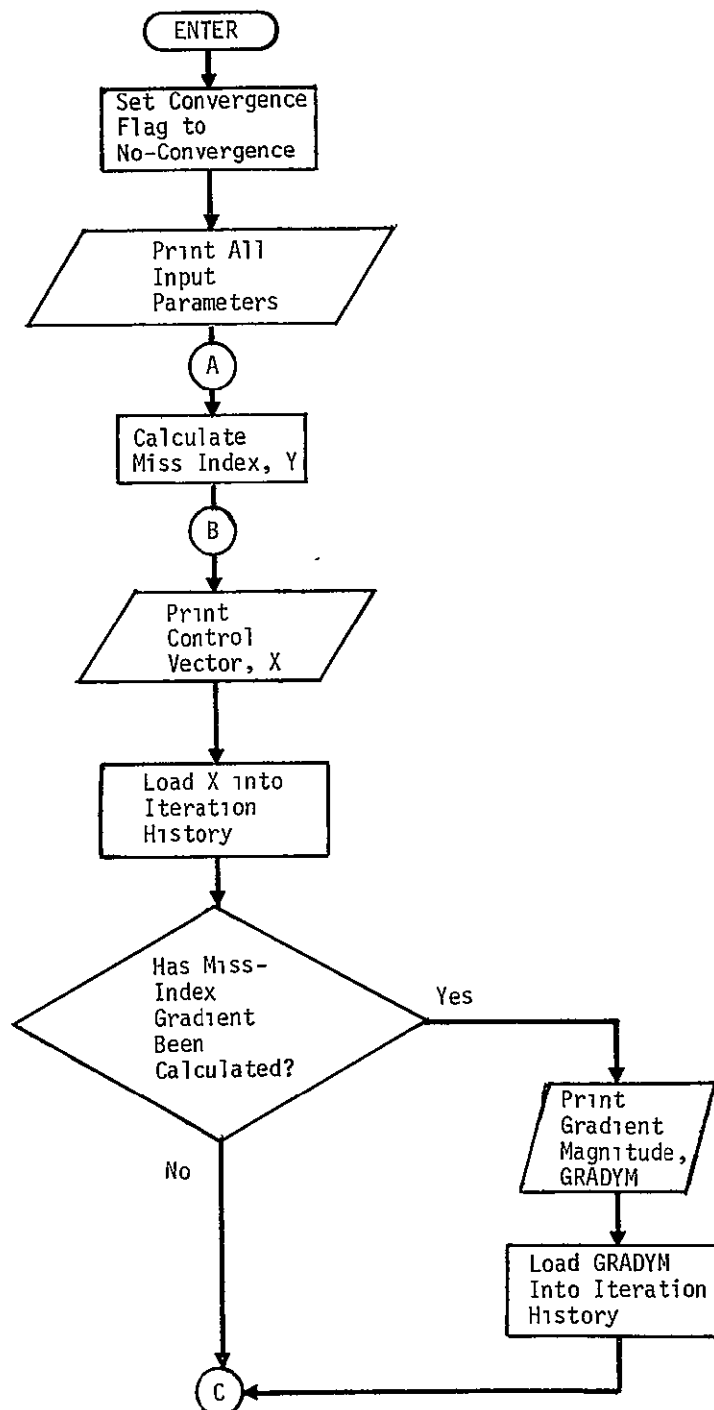
$$C_1 ||\Delta \underline{x}|| + C_2 |\Delta y| \leq \epsilon \quad (14)$$

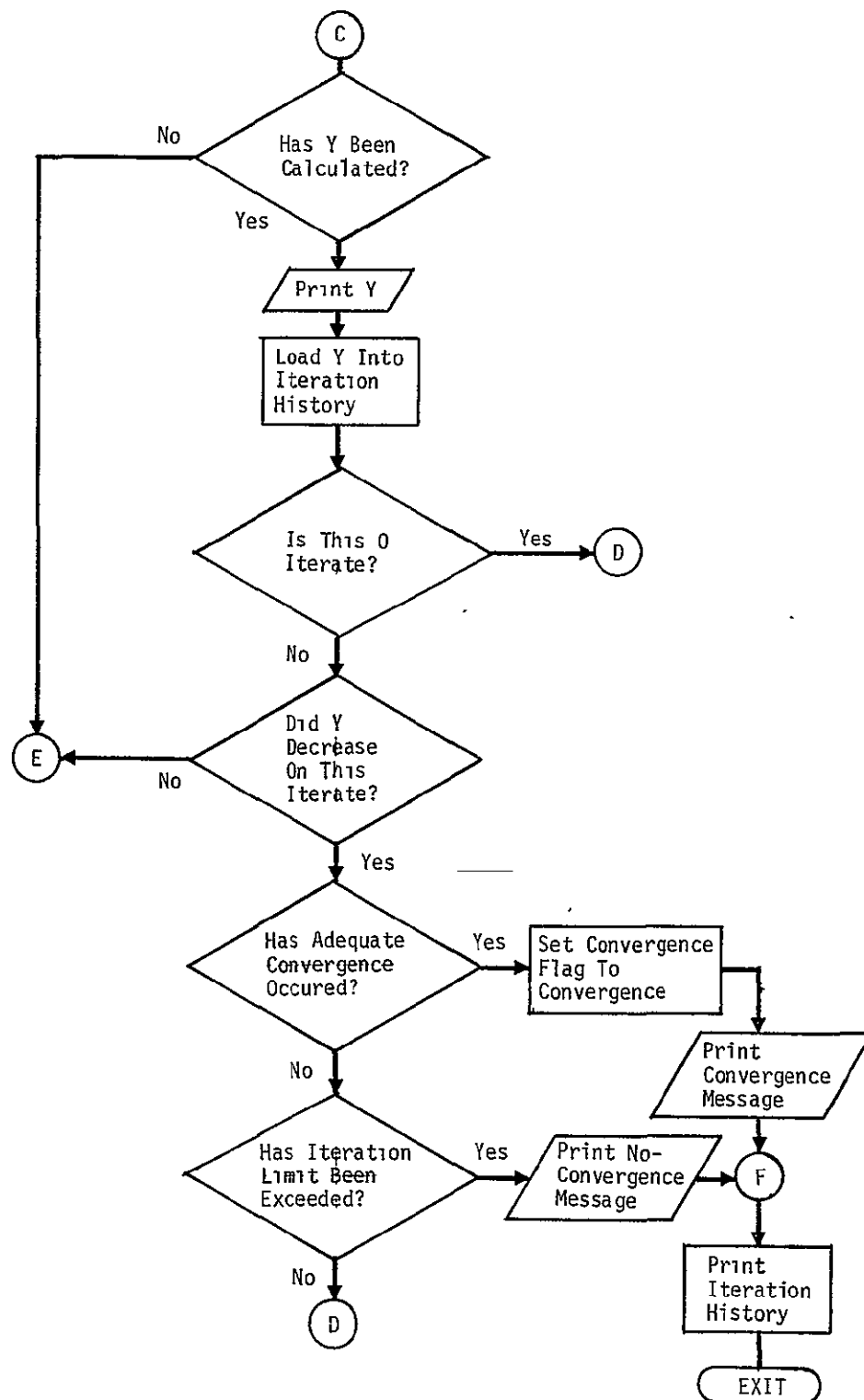
The user must supply C_1 , C_2 , and ϵ as input parameters to GAUSLS. To expedite convergence, the user should scale the components of \underline{x} so they are, as far as possible, all of the same order of magnitude, say in the range from 0.1 to 10. This scaling makes meaningful the use of a single perturbation size δ for all components of \underline{x} in approximating J and avoids numerical problems in matrix inversion and search direction calculation. Further he must supply as an input parameter the maximum number of iterations k_{\max} he will allow before terminating the algorithm.

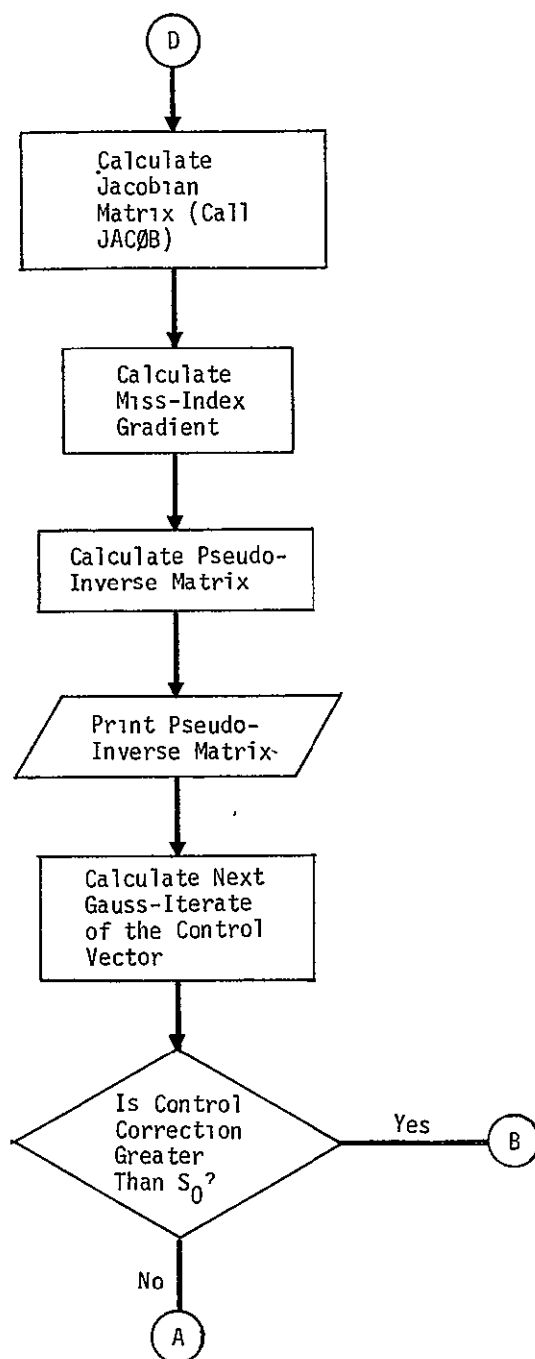
GAUSLS supplies enough output to adequately monitor either mode of the iterative least-squares process. Initially under the heading "Gauss Least-Squares Routine," it prints out all of the input parameters. These include n , m , δ , C_1 , C_2 , ϵ , s_0 , and k_{\max} . Next the user-supplied initial-control estimate \underline{x}_0 , together with the corresponding miss index $y(\underline{x}_0)$, are printed out under the heading "Gauss Iteration Point." Then the printout relative to the general k th iterate begins. All data concerning the Jacobian matrix J are printed from the subroutine JACØB under the heading "Jacobian Matrix Routine." Each iterate, of course, starts with a Jacobian matrix computation even if it eventually ends in a steepest-descent step. All of the control vectors and corresponding constraint vectors that go into the approximation of the Jacobian matrix are printed under the heading "Nominal and Perturbed Function Values." The divided-difference approximation to J is then printed under the heading "Jacobian Matrix." Next GAUSLS prints out the Gauss pseudo inverse matrix, $(J^T J)^{-1} J^T$, under the heading "Projection Matrix." Finally the next Gauss control vector iterate, the corresponding miss index, and the gradient magnitude of the previous iterate are printed out under the heading "Gauss Iteration Point." If the length of the control correction $\Delta \underline{x}$ exceeds s_0 , however, the miss index is neither calculated nor printed.

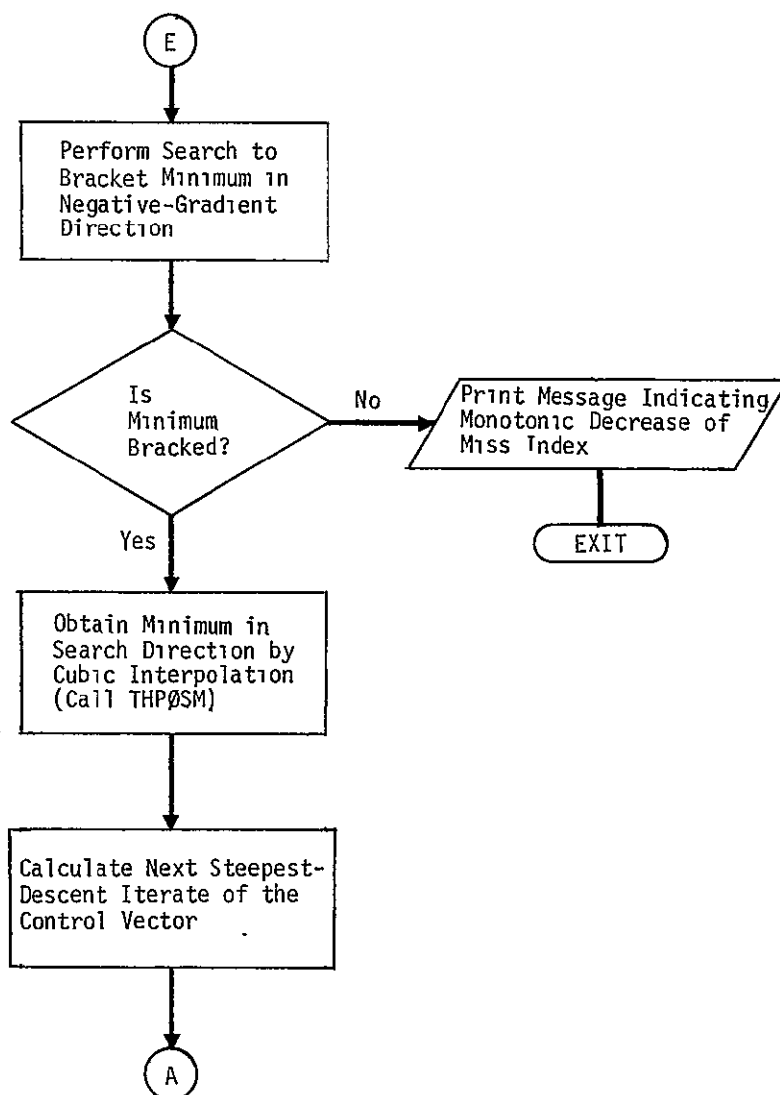
If the Gauss iterate is such that a steepest-descent step is required, GAUSLS prints out all of the pertinent data. Under the heading "Cubic Interpolation Routine" are printed all trial step lengths and corresponding miss indices used in bracketing the minimum, the input parameters to the routine THPØSM, and the minimum miss step length and index are returned by THPØSM. If the miss index decreases monotonically in the search direction, a message to that effect is printed out and execution of the program is stopped. Finally the steepest descent control iterate and the corresponding miss index is printed out under the heading "Best-Step Steepest-Descent Iteration Point." The iteration printout then is repeated with each successive iterate. When convergence finally occurs, the message "Adequate convergence occurred on previous step" is printed after the last iterate and the convergence flag, ICØNV1 is set to 1. If, on the other hand, convergence fails to occur in k_{\max} iterations, the message "Convergence did not occur" is supplied after the last iteration point and ICØNV1 is set to 2. After either of these two stopping conditions is reached, a summary of the iteration points is printed under the heading "Iteration History." This summary contains the control vector, the miss index, and the gradient to the miss index at each of the iterates in consecutive order.

GAUSLS Flow Chart









SUBROUTINE, GDATA

PURPOSE: TO INITIALIZE GENERALIZED COVARIANCE QUANTITIES

CALLING SEQUENCE: CALL GDATA

SUBROUTINES SUPPORTED: DATA

COMMON COMPUTED/USED:	EU	EV	EVA	EV8	EVK
	EVS	EW	EXI	EXSI	EXST
	EXT	GCUV	GCUW	GCVW	GCXSU
	GCXSUG	GCXSV	GCXSVG	GCXSW	GCXSWG
	GCXU	GCXUG	GCXV	GCXVG	GCXW
	GCXWG	GCXXS	GCXXSG	GDNCN	GMNCN
	GP	GPG	GPS	GPSG	GU
	GV	GW	IDNF	VARA	VARB
	VARK	VAR8			

COMMON USED:	CXSU	CXSV	CXU	CXV	CXXS
	DNCN	IDNF	MNCN	NDIM1	NDIM2
	NDIM3	NDIM4	P	PS	SIGALP
	SIGBET	SIGPRO	SIGRES	TG	U0
	V0	ZERO			

SUBROUTINE GENGID

PURPOSE: TO GENERATE THE ENSEMBLE STATISTICS OF THE ACTUAL
COMMANDED VELOCITY CORRECTION, THE ACTUAL EXECUTION
ERROR AND THE ACTUAL TARGET MISS

CALLING SEQUENCE: CALL GENGID

SUBROUTINES SUPPORTED: ERRANN

SUBROUTINES REQUIRED: SAVMAT DYN0 GNAVM MEAN MOMENT
EIGHY GQCOMP ATCEGV JACOBI

LOCAL SYMBOLS: AMAX INTERMEDIATE VARIABLE
ATC ACTUAL TARGET CONDITION 2ND MOMENT MATRIX
B INTERMEDIATE VARIABLE
BBBB BLANK LABEL ARRAY
C INTERMEDIATE VARIABLE
DELTM TIME DIFFERENCE
EBDVB MAGNITUDE OF ACTUAL STATISTICAL DELTA-V
EDVN MEAN OF ACTUAL COMMANDED VELOCITY
CORRECTION
EGM MAGNITUDE OF EIGENVECTOR CORRESPONDING TO
MAXIMUM EIGENVALUE
EGVCT EIGENVECTOR ARRAY
EGVL EIGENVALUE VECTOR
ELAB LABEL
EXTS STORAGE FOR EXI
EXSIS STORAGE FOR EXSI
EXTS STORAGE FOR EXT
EXV ACTUAL STATISTICAL DELTA-V
GAP ACTUAL VELOCITY CORRECTION 2ND MOMENT
MATRIX
GPSAVE STORAGE FOR GP
GSAVE STORAGE FOR ACTUAL CONTROL 2ND MOMENT

C.6.

MATRICES

GSAVE2 STORAGE FOR ACTUAL CONTROL 2ND MOMENT
 MATRICES

 GTG TIME OF ACTUAL GUIDANCE EVENT

 IFLAG =1 BEFORE GUIDANCE EVENT
 =2 AFTER GUIDANCE EVENT

 III INDEX DEPENDING ON GUIDANCE EVENT TYPE

 MAP INDEX OF MAXIMUM EIGENVALUE

 PEIG INTERMEDIATE ARRAY

 Q ACTUAL EXECUTION ERROR 2ND MOMENT MATRIX

 ROW INTERMEDIATE VECTOR

 S INTERMEDIATE ARRAY

 SUM INTERMEDIATE VARIABLE

 U ACTUAL COMMANDED VELOCITY CORRECTION

 VEIG INTERMEDIATE VECTOR

 ZLAB LABEL

 ZV ACTUAL EXECUTION ERROR MEANS

 ZZ INTERMEDIATE VARIABLE

COMMON COMPUTED/USED:	DUMMYQ	EXI	EXMEAN	EXSI	EXT
	GCXSUG	GCXSVG	GCXSWG	GCXUG	GCXVG
	GCXWG	GCXXSG	GP	GPG	GPSG
	XLAB				

COMMON USED:	ADA	DVUP	EE	EEE	EU
	EV	EW	EXST	FOP	FOV
	GA	GCXSU	GCXSV	GCXSW	GCXU
	GCV	GCXW	GCXXS	GPS	GU
	GV	GW	IGP	IGUID	II
	NDIM1	NDIM2	NDIM3	NDIM4	PI
	QPR	RPR	TEVN	TG	TINJ
	XIG	XSL	XU	XV	

GENGID Analysis

Subroutine GENGID controls the execution of generalized guidance events. Generalized guidance has been extended to all guidance options defined for subroutine GUIDM except for biased aimpoint guidance and impulse series thrusting.

Unlike GUIDM, which computes target dispersions and fuel budgets based on filter-generated statistics, subroutine GENGID computes target dispersions and fuel budgets based on actual statistics. In other words, the generalized covariance technique as applied to the guidance process is programmed in GENGID. The required equations are summarized below.

Before the guidance event at time t_j can be executed, it is necessary to propagate the actual control mean and control 2nd-moment matrix partitions forward to t_j from the previous guidance event at time t_{j-1} . The control mean propagates according to

$$\bar{x}_j^- = \phi \bar{x}_{j-1}^+ + \theta_{xx_s} \bar{x}_{s_o}^- + \theta_{xu} \bar{u}_o^- + \theta_{xw} \bar{w}_o^- \quad (1)$$

where ϕ , θ_{xx_s} , θ_{xu} , and θ_{xw} are state transition matrix partitions over the interval $[t_{j-1}, t_j]$, and \bar{x} , \bar{x}_s , \bar{u} , and \bar{w} denote actual position/velocity and solve-for, dynamic-consider, and ignore parameter deviation means. The notation $()^-$ indicates actual values as opposed to the unprimed assumed values, while $()^-$ and $()^+$ indicate values immediately before and after the execution of the guidance event, respectively. The actual control position/velocity 2nd-moment matrix is defined by

$$P_{c_j} = E \begin{bmatrix} \bar{x}_j^- & \bar{x}_j^{+T} \end{bmatrix}. \quad (2)$$

The remaining control 2nd-moment matrix partitions are defined similarly. The propagation equations appearing in subroutine GNAVM are used to propagate the control 2nd-moment matrix partitions over the interval $[t_{j-1}, t_j]$.

The actual target state deviation $\delta \tau_j'$ is related to the actual state deviation x_j' at time t_j according to

$$\delta \tau_j' = \eta_j x_j' \quad (3)$$

where η_j is the variation matrix for the appropriate midcourse guidance policy. The mean of $\delta \tau_j'$ is given by

$$E [\delta \tau_j'] = \eta_j E [x_j']. \quad (4)$$

The statistical target dispersions are represented by the actual target condition 2nd-moment matrix W_j' , which is defined as

$$W_j' = E [\delta \tau_j' \delta \tau_j'^T]. \quad (5)$$

Substitution of equation (3) into equation (5) yields

$$W_j' = \eta_j P_{c_j}' \eta_j^T. \quad (6)$$

Equations (4) and (6) are evaluated immediately before and after the guidance correction to determine how much the target errors have actually been reduced by the velocity correction at t_j .

The actual commanded velocity correction 2nd-moment matrix is defined by

$$S_j' = E [\hat{\Delta V}_j' \hat{\Delta V}_j'^T] \quad (7)$$

where the actual commanded velocity correction is given by

$$\hat{\Delta V}_j' = \Gamma_j \hat{x}_j' = \Gamma_j (x_j' + \tilde{x}_j'). \quad (8)$$

The guidance matrix Γ_j corresponds to the appropriate linear mid-course guidance policy. The equation used to evaluate S_j' is given by

$$S_j' = \Gamma_j (P_{c_j}' - P_{k_j}') \Gamma_j^T \quad (9)$$

where all $E \begin{bmatrix} \dot{x}_j & \dot{x}_j^T \end{bmatrix}$ terms have been neglected in the derivation of equation (9).

The mean of the actual commanded velocity correction is obtained by applying the expectation operator to equation (8):

$$E \begin{bmatrix} \dot{\Delta V}_j \end{bmatrix} = \Gamma_j \left\{ E \begin{bmatrix} \dot{x}_j \end{bmatrix} + E \begin{bmatrix} \dot{x}_j \end{bmatrix} \right\}. \quad (10)$$

Since this equation gives no useful information for fuel-sizing studies, the Hoffman-Young formula will be used to evaluate

$$E \begin{bmatrix} |\hat{\Delta V}_j| \end{bmatrix} = \sqrt{\frac{2A}{\pi}} \left(1 + \frac{B (\pi - 2)}{A^2 \sqrt{5.4}} \right) \quad (11)$$

where

$$A = \text{trace } S_j$$

$$B = \lambda_1' \lambda_2' + \lambda_1' \lambda_3' + \lambda_2' \lambda_3',$$

and λ_1' , λ_2' , and λ_3' are the eigenvalues of the 2nd-moment matrix S_j .

The actual effective or statistical ΔV is defined as

$$"E \begin{bmatrix} \hat{\Delta V}_j \end{bmatrix}" = E \begin{bmatrix} |\hat{\Delta V}_j| \end{bmatrix} \cdot \alpha_j \quad (12)$$

where α_j denotes a unit vector in the most likely direction of the velocity correction. The most likely direction is assumed to be aligned with the eigenvector associated with the maximum eigenvalue of S_j .

With $"E \begin{bmatrix} \hat{\Delta V}_j \end{bmatrix}"$ available, the actual execution error statistics can be computed (by calling subroutine GQCØMP). These are the actual execution error mean $E \begin{bmatrix} \delta \Delta V_j \end{bmatrix}$ and 2nd-moment matrix \tilde{Q}_j defined as

$$\tilde{Q}_j = E \begin{bmatrix} \delta \Delta V_j & \delta \Delta V_j^T \end{bmatrix}. \quad (13)$$

It remains to summarize the equations which are used to update all actual control and knowledge means and 2nd-moment matrix partitions immediately following the execution of a guidance event. The actual estimation error means and 2nd-moment matrix partitions are updated using the following equations:

$$E \begin{bmatrix} \tilde{x}_j^+ \end{bmatrix} = E \begin{bmatrix} \tilde{x}_j^- \end{bmatrix} - A \cdot E \begin{bmatrix} \delta \Delta V_j \end{bmatrix} \quad (14)$$

$$E \begin{bmatrix} \tilde{x}_{s_j}^+ \end{bmatrix} = E \begin{bmatrix} \tilde{x}_{s_j}^- \end{bmatrix} \quad (15)$$

$$P_{k_j}^+ = P_{k_j}^- + A Q_j A^T - A \cdot E \begin{bmatrix} \delta \Delta V_j \end{bmatrix} \cdot E \begin{bmatrix} \tilde{x}_j^- \tilde{x}_j^{T-} \end{bmatrix} - E \begin{bmatrix} \tilde{x}_j^- \end{bmatrix} \cdot E \begin{bmatrix} \delta \Delta V_j \end{bmatrix} \cdot A^T \quad (16)$$

$$P_{s_{k_j}}^+ = P_{s_{k_j}}^- \quad (17)$$

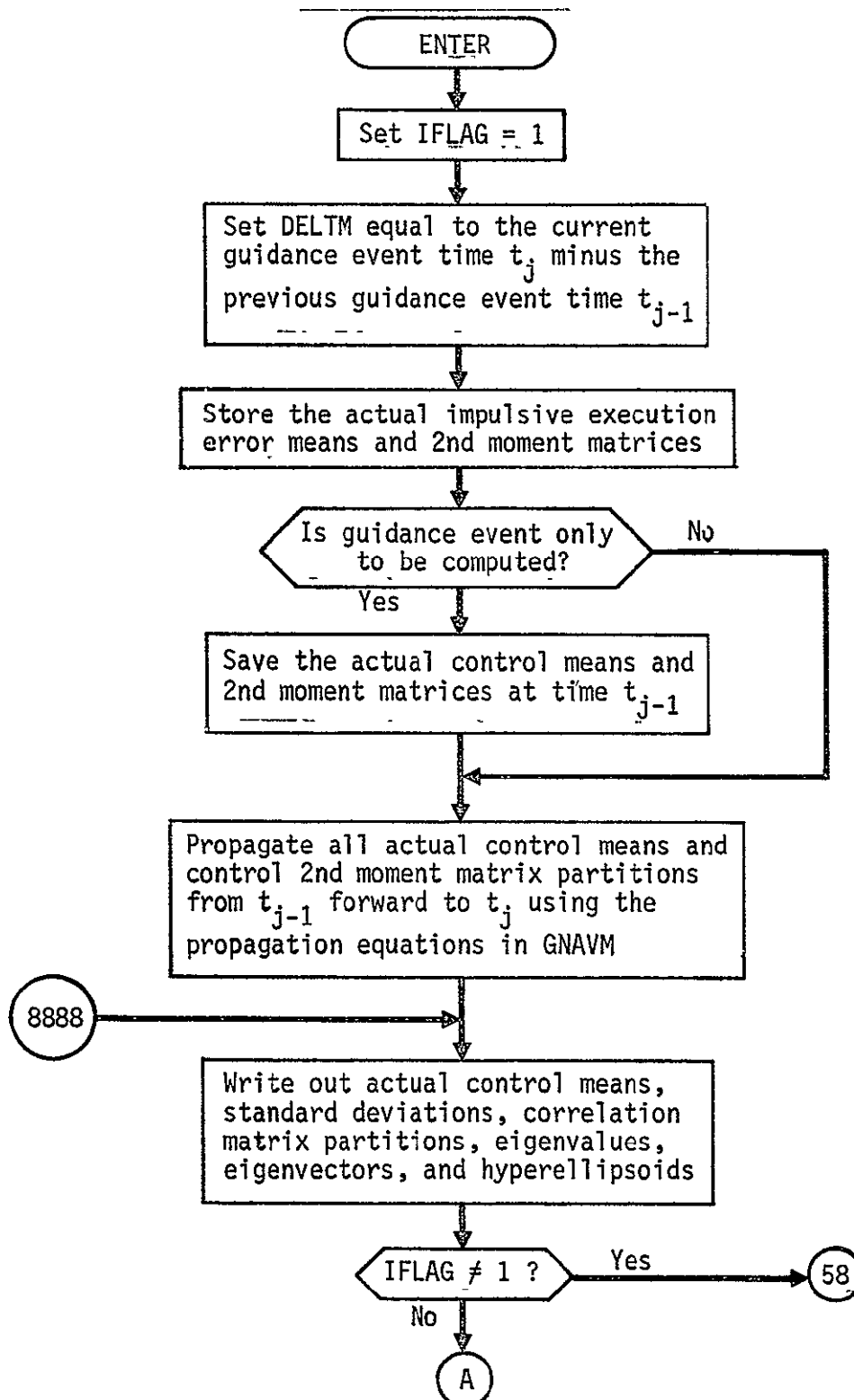
where $A = [0 \mid I]^T$. The actual deviation means are updated using the following equations:

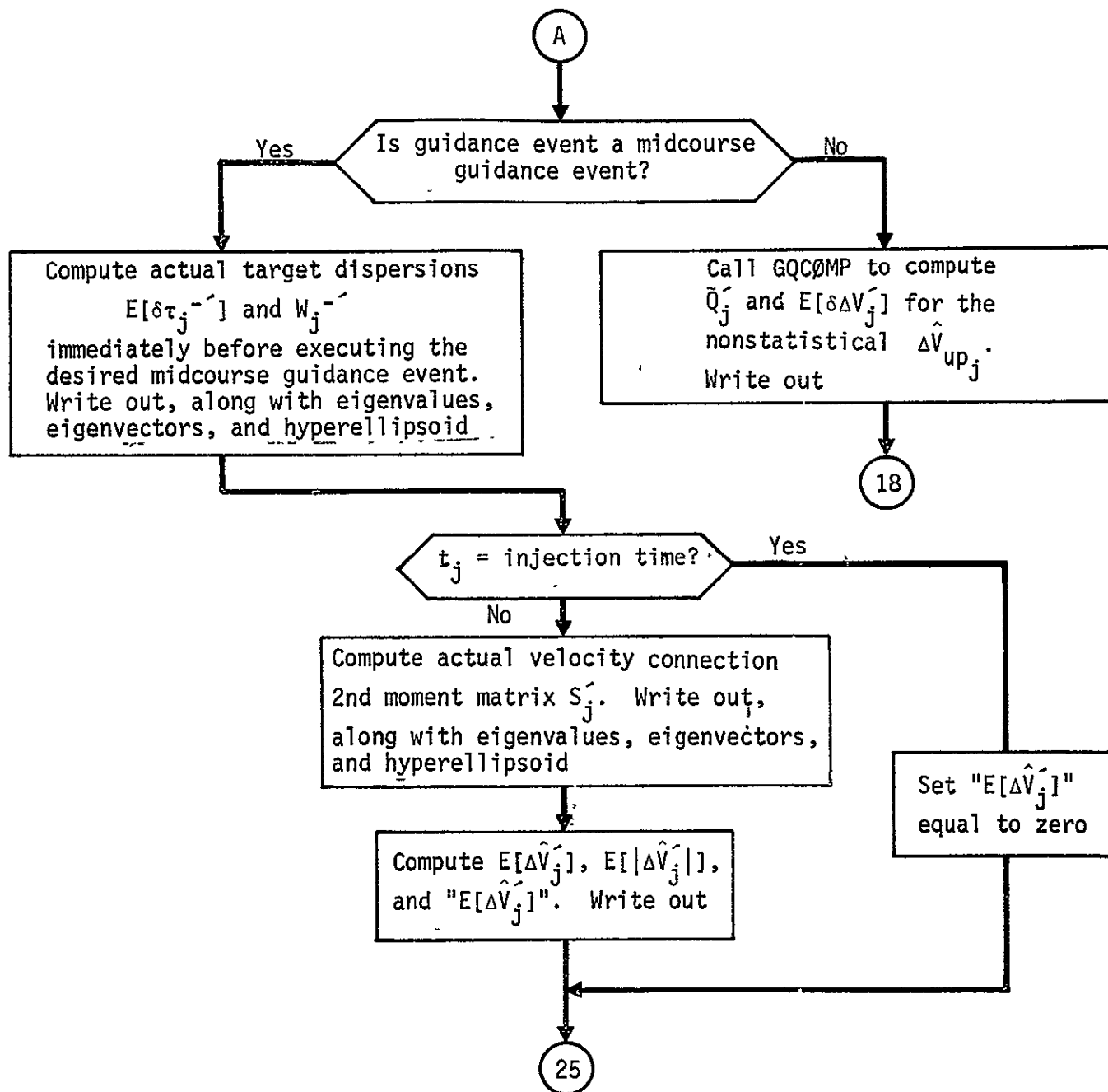
$$E \begin{bmatrix} \tilde{x}_j^+ \end{bmatrix} = - E \begin{bmatrix} \tilde{x}_j^+ \end{bmatrix} \quad (18)$$

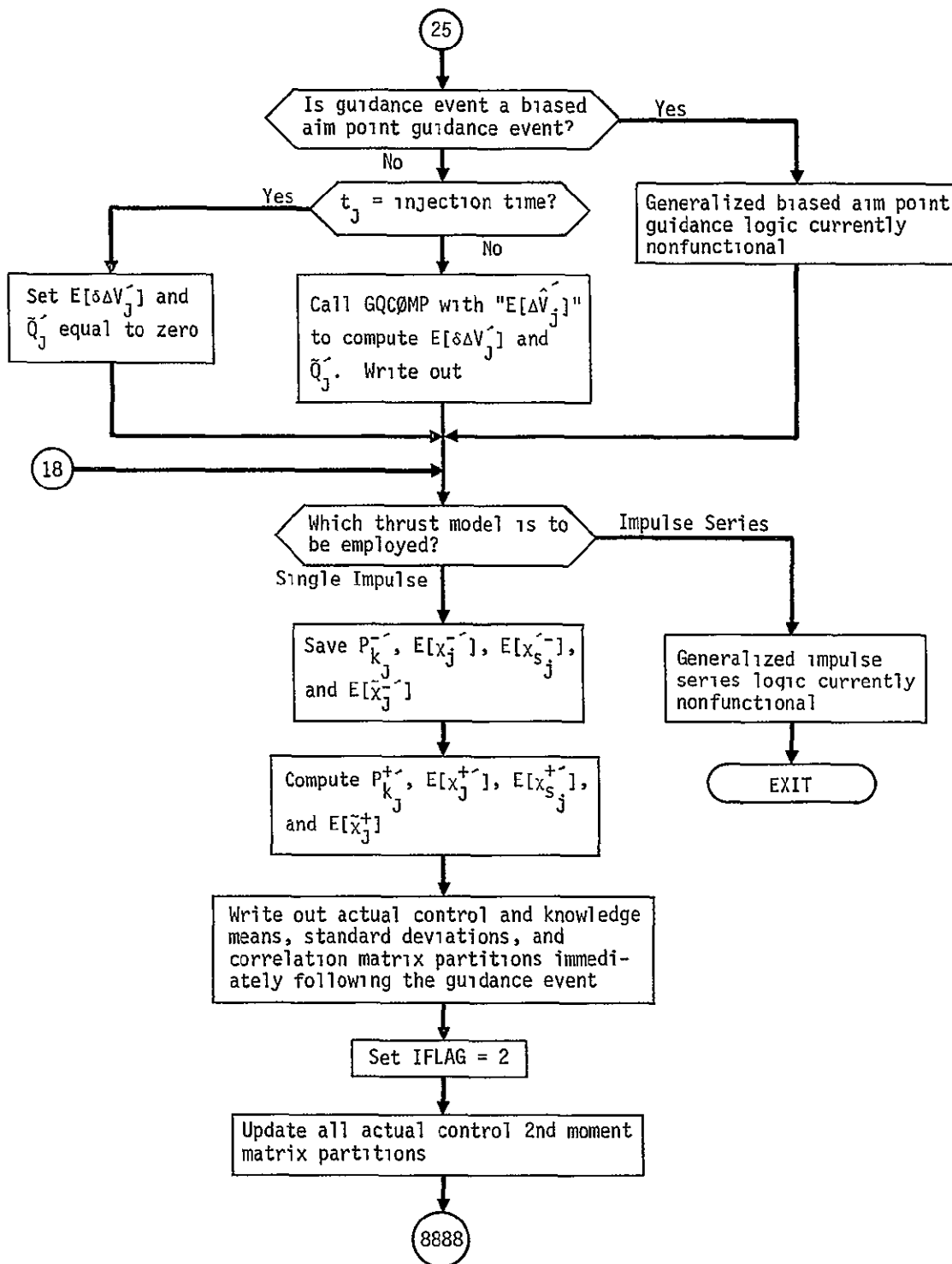
$$E \begin{bmatrix} \tilde{x}_{s_j}^+ \end{bmatrix} = - E \begin{bmatrix} \tilde{x}_{s_j}^+ \end{bmatrix} \quad (19)$$

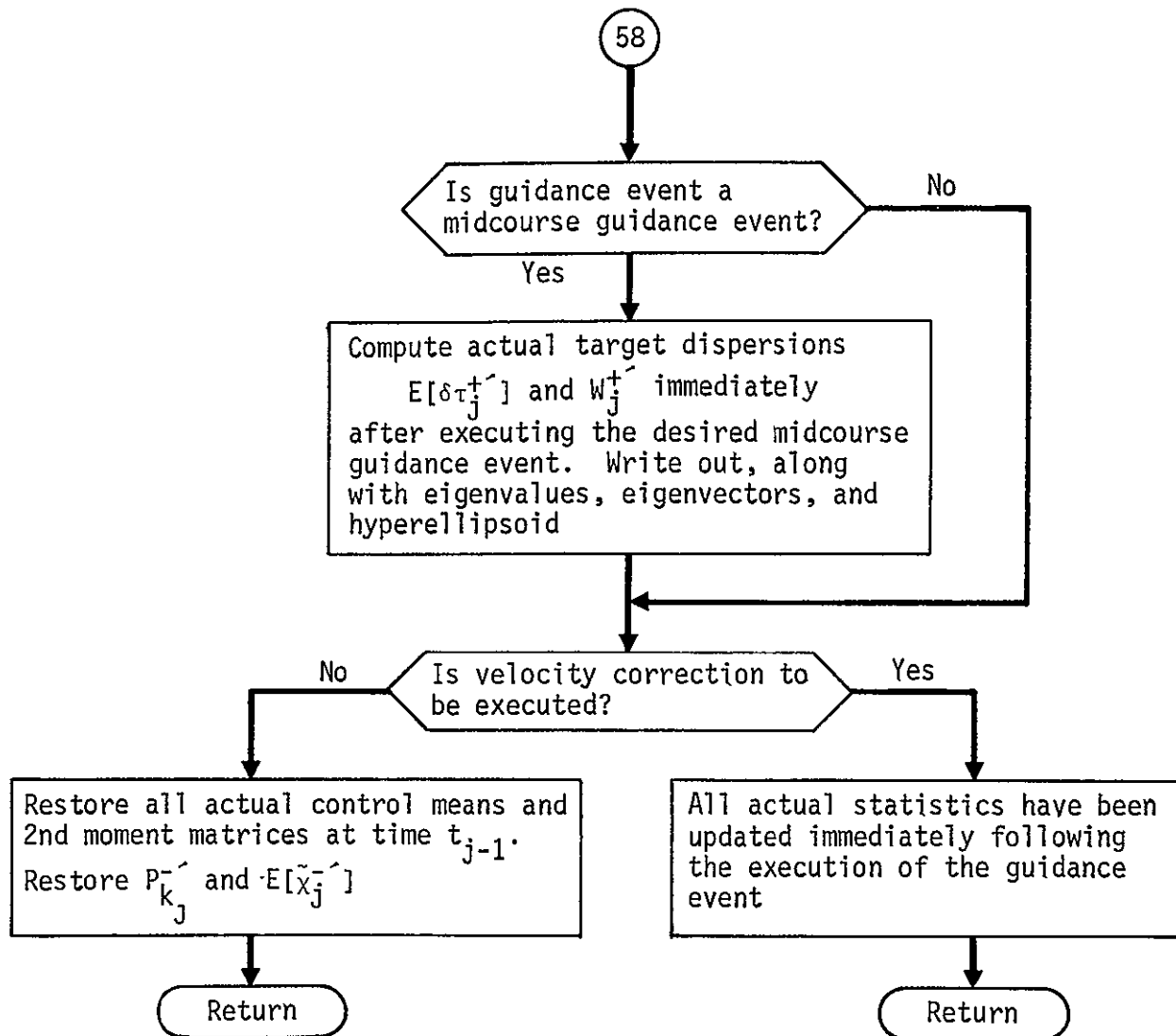
The entire set of actual control 2nd-moment matrix partitions is updated by equating them to the corresponding actual knowledge 2nd-moment matrix partitions at t_j^+ .

GENGID Flow Chart









SUBROUTINE GHA

PURPOSE: TO COMPUTE THE GREENWICH HOUR ANGLE AND THE UNIVERSAL TIME (IN DAYS) WHICH IS USED IN THE TRACKING MODULE TO ORIENT THE TRACKING STATIONS ON A SPHERICAL ROTATING EARTH.

CALLING SEQUENCE: CALL GHA

ARGUMENTS: NONE

SUBROUTINES SUPPORTED: DATA1S DATA1

LOCAL SYMBOLS:

D	NUMBER OF DAYS IN TSTAR
EQMEG	EARTH ROTATION RATE
GH	GREENWICH HOUR ANGLE
ID	INTERMEDIATE VARIABLE
REFJD	JULIAN DATE OF JAN. 0, 1950
TFRAC	FRACTION OF DAY IN TSTAR
TSTAR	JULIAN DATE, EPOCH JAN. 0, 1950, OF INITIAL TRAJECTORY TIME

COMMON COMPUTED: UNIVT

COMMON USED: DATEJ EM13

GHA Analysis

Subroutine GHA computes the Greenwich hour angle in degrees and days at some epoch T^* referenced to 1950 January 1^d0^h. Epoch T^* is computed from

$$T^* = J.D._0 + 2415020.0 - J.D._{REF}$$

where

$J.D._0$ = Julian date at launch time t_0 referenced to 1900 January 0^d12^h.

$J.D._{REF}$ = Reference Julian date 2433282.5

= 1950 January 1^d0^h referenced to January 0^d12^h of the year 4713 B.C.

and 2415020.0 = 1900 January 0^d12^h referenced to January 0^d12^h of the year 4713 B.C.

Then T^* is the Julian date at launch time t_0 referenced to 1950 January 1^d0^h.

The Greenwich hour angle corresponding to T^* is given by _____

$$GHA(T^*) = 100.0755426 + 0.985647346d + 2.9015 \times 10^{-13} d^2 + \omega t$$

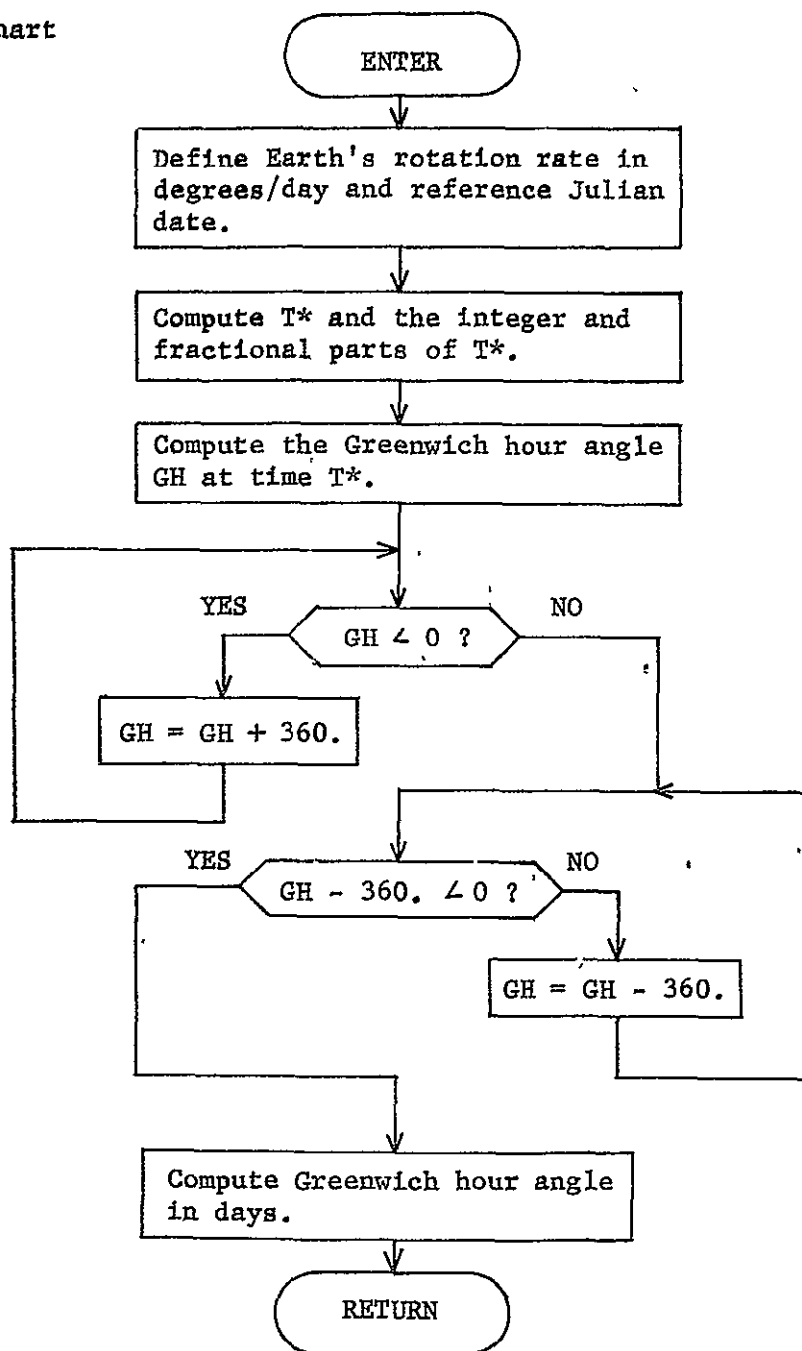
where $0 \leq GHA(T^*) < 360^\circ$

and d = integer part of T^* , t = fractional part of T^* ,

and ω = Earth's rotation rate in degrees/day.

The Greenwich hour angle in days is given by $\frac{GHA}{\omega}$.

GHA Flow Chart



SUBROUTINE GIDANS

PURPOSE: EXECUTIVE ROUTINE FOR COMPUTATION OF REQUIRED GUIDANCE
EVENT

CALLING SEQUENCE: CALL GIDANS

SUBROUTINES SUPPORTED: NOMNAL NONLIN

SUBROUTINES REQUIRED: EXECUTE ZERIT TARGET INSERTS TIME
VMP

LOCAL SYMBOLS: DTIME DELTA TIME (DAYS) BETWEEN ORBIT INSERTION
COMPUTATION AND EXECUTION

IZER VECTOR OF CODES USED FOR RETARGETING
IZER(KUR)=0, DO NOT RECOMPUTE ZERO ITERATE
#0, RECOMPUTE ZERO ITERATE

I INDEX

KTYPE VALUE OF KTYPE(KUR) INDICATING TYPE OF
EVENT
=1, ORIGINAL TARGETING
=2, RETARGETING
=3, ORBIT INSERTION

MODEL DOES NOT APPEAR IN CURRENT VERSION SEE
EXECUTE

COMMON COMPUTED/USED: KMXQ KWHIT TIMG

COMMON COMPUTED: DELV KTIM ZDAT

COMMON USED: DELTAV KTYPE KUR MDL RIN

GIDANS Analysis

GIDANS is an executive routine responsible for processing a guidance maneuver for the computation of the velocity increment Δv to the execution of the correction.

Before entry to GIDANS, TRJTRY has computed the index of the current event (KUR) and has integrated the nominal trajectory to the time of the event. GIDANS now evaluates the KUR component of two integer arrays -- KTYP and KMXQ. The values of these flags determine the operation of GIDANS. The flag KTYP specifies the type of guidance event to be performed, while KMXQ prescribes the compute/execute mode to be used according to

```

KTYP = -1  Termination event
        1  Targeting event
        2  Retargeting event
        3  Orbit insertion
        4  Main probe propagation
        5  Miniprobe targeting

KMXQ = 1  Compute  $\Delta \vec{y}$  only
        2  Execute  $\Delta v$  only
        3  Compute and execute  $\Delta \vec{y}$ 
        4  Compute but execute  $\Delta v$  later
    
```

GIDANS first checks for a termination event. If the current index prescribes such an event, the flag KWIT is set to 1 and a return is made to the main program NOMNAL.

In preparation for a normal guidance event, GIDANS calls VMP with the current spacecraft heliocentric state and a time increment of zero to restore the F and V arrays providing the current geometry of spacecraft and planets. If the current event is an execute-only mode, the transfer is made to the execution section of GIDANS for the addition of the preset velocity increment.

Otherwise GIDANS interrogates KTYP for the type of maneuver to be computed. For a targeting event, subroutine TARGET is called directly for the computation of the Δv necessary to satisfy input target conditions. After calling TARGET, the F and V arrays are restored as indicated above.

A retargeting event is defined as a targeting event that requires computation of a new zero iterate. Thus a retargeting event is an event in which the current nominal state when integrated forward would miss the target conditions badly. Such an event would be the broken-plane correction. For this event TRJTRY stores the current position (and possibly the target position) in the ZDAT array. It then calls ZERIT for the computation of the massless-planet's initial velocity consistent with the target conditions. It then operates identically to the targeting event.

The third guidance maneuver is the insertion event. GIDANS calls INSERS for the computation of the velocity increment Δv and the time interval Δt before it is to be executed.

The main-probe propagation event involves storing the current spacecraft state, propagating the main probe to an appropriate stopping condition while printing a time history, and restoring the original state in preparation for the next event. It is carried out in a single call to the subroutine MPPROP. Upon return to GIDANS, the F and V arrays are restored as indicated above.

The miniprobe targeting event, although somewhat complicated, is completely executed by the single subroutine TPRTG. The current bus state is first stored. Next the miniprobe release controls are calculated to apply at the current time to target three miniprobes respectively to three target sites characterized by input values of declination and right ascension. Using the minimum-miss release controls, each miniprobe is then propagated from release to a stopping condition while a time history is concurrently printed. Finally, the original bus state is restored. On returning to GIDANS, the F and V arrays are restored as usual.

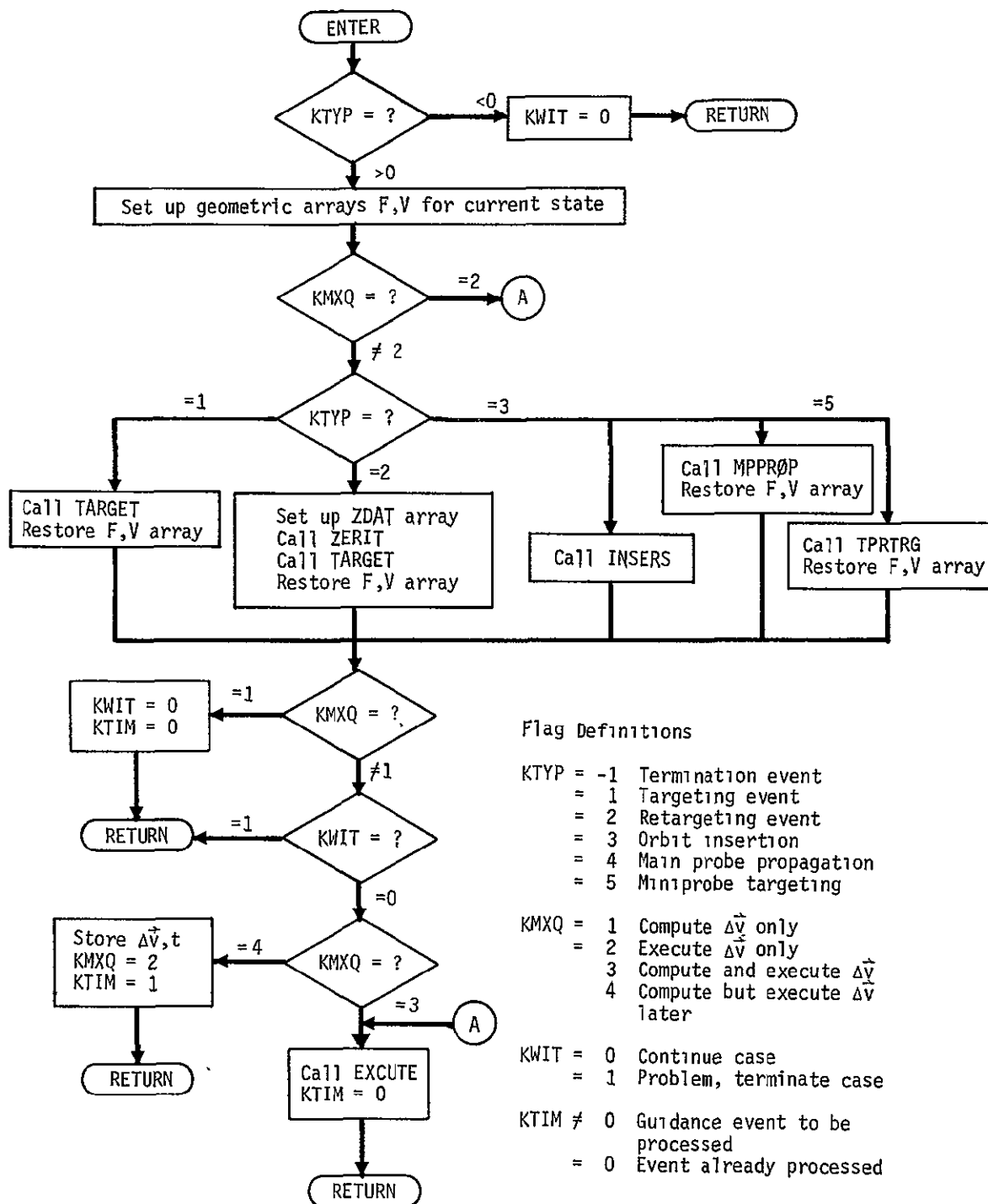
The three subroutines TARGET, INSERS, and TPRTG signal trouble to GIDANS via the flag KWIT. If problems are encountered in their execution, e.g., failure to converge in TARGET or TPRTG or the impossibility of insertion in INSERS, KWIT is set to 1. Otherwise KWIT = 0. On return to NOMNAL, if KWIT = 1 the current case is terminated while if KWIT = 0 it is continued.

If the current event is a compute-only mode, TRJTRY now sets KWIT = 0 (so that the program will continue regardless of whether the correction computations were successful) and returns to NOMNAL. However if the current event failed (KWIT = 1) and was to be executed (KMXQ \neq 1) GIDANS considers this a fatal error for the current case and returns with KWIT = 1.

If the compute/execute mode is compute-execute later ($KMXQ = 4$) as is the insertion event, GIDANS now sets up for the subsequent execute-only event. The Δv computed is stored in the DELV array, the time of the execution is computed ($t_{ex} = t_k + \Delta t$) and stored in the TIMG array, and the KMXQ flag is set to a 2 (execute-only). The return is then made to NOMINAL.

For an event to be executed at the current time ($KMXQ = 2,3$), GIDANS now calls EXECUTE for the completion of that task.

It should be noted that for all events that are completed at this time, the KUR components of the KTIM array are set equal to 0 so they are no longer considered in determining the next event in TRJTRY. Only in the case of $KMXQ = 4$ is the KTIM flag nonzero on exit from GIDANS.



SUBROUTINE GNAVM

PURPOSE: TO PROPAGATE ASSUMED COVARIANCE MATRIX PARTITIONS P, CXXS, CXU, CXV, PS, CXSU, CXSV, OR ACTUAL SECOND MOMENT MATRIX PARTITIONS GP, GCXXS, GCXU, GCXV, GCXW, GPS, GCXSU, GCXSV, GCXSW FROM THE TIME OF THE LAST MEASUREMENT OR EVENT TO THE PRESENT TIME AND TO UPDATE THESE MATRIX PARTITIONS IF A MEASUREMENT IS TO BE PROCESSED

CALLING SEQUENCE: CALL GNAVM(NR, IFLAG1, ICODE, U0, V0, GCXW, GCXSW, P, CXXS, CXU, CXV, PS, CXSU, CXSV, Q, R)

ARGUMENTS:	NR	I NUMBER OF ROWS IN THE OBSERVATION MATRIX
	IFLAG1	I =1 FOR ASSUMED COVARIANCE PROCESSING =2 FOR ACTUAL SECOND MOMENT PROCESSING
	ICODE	I =0 FOR UPDATE =1 FOR PROPAGATION
	U0	I ACTUAL OR ASSUMED DYNAMIC CONSIDER PARAMETER 2ND MOMENT MATRIX
	V0	I ACTUAL OR ASSUMED MEASUREMENT CONSIDER PARAMETER 2ND MOMENT MATRIX
	GCXW	I ACTUAL POSITION-VELOCITY STATE / IGNORE PARAMETER 2ND MOMENT MATRIX
	GCXSW	I ACTUAL SOLVE-FOR PARAMETER / IGNORE PARAMETER 2ND MOMENT MATRIX
	P	I ACTUAL OR ASSUMED POSITION-VELOCITY 2ND MOMENT MATRIX 2ND MOMENT MATRIX
	CXXS	I ASSUMED OR ACTUAL POSITION-VELOCITY STATE / SOLVE-FOR PARAMETER 2ND MOMENT MATRIX
	CXU	I ASSUMED OR ACTUAL POSITION-VELOCITY STATE / DYNAMIC CONSIDER PARAMETER 2ND MOMENT MATRIX
	CXV	I ASSUMED OR ACTUAL POSITION-VELOCITY STATE / MEASUREMENT CONSIDER PARAMETER 2ND MOMENT MATRIX
	PS	I ASSUMED OR ACTUAL SOLVE-FOR PARAMETER COVARIANCE OR 2ND MOMENT MATRIX
	CXSU	I ASSUMED OR ACTUAL SOLVE-FOR PARAMETER / DYNAMIC CONSIDER PARAMETER 2ND MOMENT MATRIX

CXSV I ASSUMED OR ACTUAL SOLVE-FOR PARAMETER
 / MEASUREMENT CONSIDER PARAMETER 2ND
 MOMENT MATRIX

Q I ASSUMED OR ACTUAL DYNAMIC NOISE 2ND MOMENT
 MATRIX

R I ASSUMED OR ACTUAL MEASUREMENT NOISE
 2ND MOMENT MATRIX

SUBROUTINES SUPPORTED: ERRANN SETEVN GUIDM PRED GENGD PROBE

SUBROUTINES REQUIRED: GAIN1 GAIN2

LOCAL SYMBOLS: AKW INTERMEDIATE ARRAY

 DS INTERMEDIATE ARRAY

 ES INTERMEDIATE ARRAY

 FS INTERMEDIATE ARRAY

 IEND NR-1

 NDIM4S NDIM4 VALUE STORAGE

 N1 NDIM1-1

 SUM INTERMEDIATE VARIABLE

 SW INTERMEDIATE ARRAY

COMMON COMPUTED/USED: AK AL AM AN CXSUP

 CXSVP CXUP CXVP CXXSP G

 GCUV GCUW GCVW GCXSWP GCXWP

 HPHR IGAIN GW H HALF

 JPR ZERO PP PSP S

COMMON USED: NDIM1 NDIM2 NDIM3 NDIM4 PHI

 TXU TXW TXXS

GNAVM Analysis

Subroutine GNAVM propagates and updates (at a measurement) both assumed (or filter) covariance matrix partitions and actual 2nd moment matrix partitions. The equations programmed in GNAVM are independent of the filter algorithm employed to generate gain matrices.

The covariance and 2nd moment matrix partitions manipulated by GNAVM are defined as follows:

$$\begin{aligned}
 P &= E[\tilde{x} \tilde{x}^T] & P_s &= E\left[\tilde{x}_s \tilde{x}_s^T\right] \\
 C_{xx_s} &= E\left[\tilde{x} \tilde{x}_s^T\right] & C_{x_s u} &= E[\tilde{x}_s \tilde{u}^T] \\
 C_{xu} &= E[\tilde{x} \tilde{u}^T] & C_{x_s v} &= E[\tilde{x}_s \tilde{v}^T] \\
 C_{xv} &= E[\tilde{x} \tilde{v}^T] & C_{x_s w} &= E[\tilde{x}_s \tilde{w}^T] \\
 C_{xw} &= E[\tilde{x} \tilde{w}^T]
 \end{aligned} \tag{1}$$

The following matrix partitions are used in GNAVM, but are not changed in GNAVM:

$$\begin{aligned}
 C_{uv} &= E[\tilde{u} \tilde{v}^T] \\
 C_{uw} &= E[\tilde{u} \tilde{w}^T] \\
 C_{vw} &= E[\tilde{v} \tilde{w}^T] \\
 U &= E[\tilde{u} \tilde{u}^T] \\
 V &= E[\tilde{v} \tilde{v}^T] \\
 W &= E[\tilde{w} \tilde{w}^T]
 \end{aligned} \tag{2}$$

In these definitions \tilde{x} , \tilde{x}_s , \tilde{u} , \tilde{v} , and \tilde{w} represent, respectively, the estimation errors in position/velocity state, solve-for parameters, dynamic consider parameters, measurement consider parameters, and ignore parameters. Ignore parameters, of course, are not defined when assumed (or filter) covariance matrix partitions are being propagated or updated. Furthermore, the assumed C_{uv} has been set to zero.

The equations used to propagate covariances or 2nd moment matrices from time t_k to t_{k+1} are summarized:

$$P_{k+1}^- = \left(\phi P_k^+ + \theta_{xx_s} C_{xx_s}^{+T} + \theta_{xu} C_{xu_k}^{+T} + \theta_{xw} C_{xw_k}^{+T} \right) \phi^T + C_{xx_s}^- \theta_{xx_s}^T + C_{xu_{k+1}}^- \theta_{xu}^T + C_{xw_{k+1}}^- \theta_{xw}^T + Q_{k+1} \quad (3)$$

$$C_{xx_s}^- = \phi C_{xx_s}^+ + \theta_{xx_s} P_{s_k}^+ + \theta_{xu} C_{x_s u_k}^{+T} + \theta_{xw} C_{x_s w_k}^{+T} \quad (4)$$

$$C_{xu_{k+1}}^- = \phi C_{xu_k}^+ + \theta_{xx_s} C_{x_s u_k}^+ + \theta_{xu} U_o + \theta_{xw} C_{uw_o}^T \quad (5)$$

$$C_{xv_{k+1}}^- = \phi C_{xv_k}^+ + \theta_{xx_s} C_{x_s v_k}^+ + \theta_{xu} C_{uv_o} + \theta_{xw} C_{vw_o}^T \quad (6)$$

$$C_{xw_{k+1}}^- = \phi C_{xw_k}^+ + \theta_{xx_s} C_{x_s w_k}^+ + \theta_{xu} C_{uw_o} + \theta_{xw} W_o \quad (7)$$

$$P_{s_{k+1}}^- = P_{s_k}^+ \quad (8)$$

$$C_{x_s u_{k+1}}^- = C_{x_s u_k}^+ \quad (9)$$

$$C_{x_s v_{k+1}}^- = C_{x_s v_k}^+ \quad (10)$$

$$C_{x_s w_{k+1}}^- = C_{x_s w_k}^+ \quad (11)$$

In these equations $()^-$ indicates immediately prior to processing a measurement; $()^+$, immediately after. The state transition matrices over the interval $[t_k, t_{k+1}]$ are indicated by Φ , θ_{xx_s} , θ_{xu} , and θ_{xw} . The dynamic noise covariance or 2nd moment matrix is denoted by Q_{k+1} .

Before covariance (or 2nd moment) matrix partitions can be updated at a measurement, the measurement residual covariance (or 2nd moment) matrix, defined by

$$J_{k+1} = E \begin{bmatrix} \epsilon_{k+1} & \epsilon_{k+1}^T \end{bmatrix} \quad (12)$$

must be computed. The required equations are summarized

$$J_{k+1} = HA_{k+1} + MB_{k+1} + GD_{k+1} + LE_{k+1} + NF_{k+1} + R_{k+1} \quad (13)$$

$$A_{k+1} = P_{k+1}^- H^T + C_{xx_s k+1}^- M^T + C_{xu k+1}^- G^T + C_{xv k+1}^- L^T + C_{xw k+1}^- N^T \quad (14)$$

$$B_{k+1} = P_{s k+1}^- M^T + C_{xx_s k+1}^- H^T + C_{x_s u k+1}^- G^T + C_{x_s v k+1}^- L^T + C_{x_s w k+1}^- N^T \quad (15)$$

$$D_{k+1} = C_{xu k+1}^- H^T + C_{x_s u k+1}^- M^T + U_o G^T + C_{uw_o}^- N^T + C_{uv_o}^- L^T \quad (16)$$

$$E_{k+1} = C_{xv k+1}^- H^T + C_{x_s v k+1}^- M^T + C_{vw_o}^- N^T + V_o L^T + C_{uv_o}^- G^T \quad (17)$$

$$1 - \gamma_i = \omega_i$$

$$F_{k+1} = W_o N^T + C_{xw_{k+1}}^{-T} H^T + C_{x_s w_{k+1}}^{-T} M^T + C_{vw_o}^{-T} L^T + C_{uw_o}^{-T} G^T \quad (18)$$

In these equations H , M , G , L , and N represent observation matrix partitions, and R_{k+1} represents the measurement noise covariance (or 2nd moment) matrix.

Gain matrices K_{k+1} and S_{k+1} are also required before covariance (or 2nd moment) matrix partitions can be updated. These are not computed in GNAVM but are obtained by calling either subroutine GAIN1 or GAIN2, depending on which recursive estimation algorithm is desired.

With J_{k+1} , K_{k+1} , and S_{k+1} available, the following equations are used in the updating process:

$$P_{k+1}^+ = P_{k+1}^- - K_{k+1} A^T - A K_{k+1}^T + K_{k+1} J_{k+1} K_{k+1}^T \quad (19)$$

$$C_{xx_s k+1}^+ = C_{xx_s k+1}^- - K_{k+1} B^T - A S_{k+1}^T + K_{k+1} J_{k+1} S_{k+1}^T \quad (20)$$

$$C_{xu_{k+1}}^+ = C_{xu_{k+1}}^- - K_{k+1} D^T \quad (21)$$

$$C_{xv_{k+1}}^+ = C_{xv_{k+1}}^- - K_{k+1} D^T \quad (22)$$

$$C_{xw_{k+1}}^+ = C_{xw_{k+1}}^- - K_{k+1} F^T \quad (23)$$

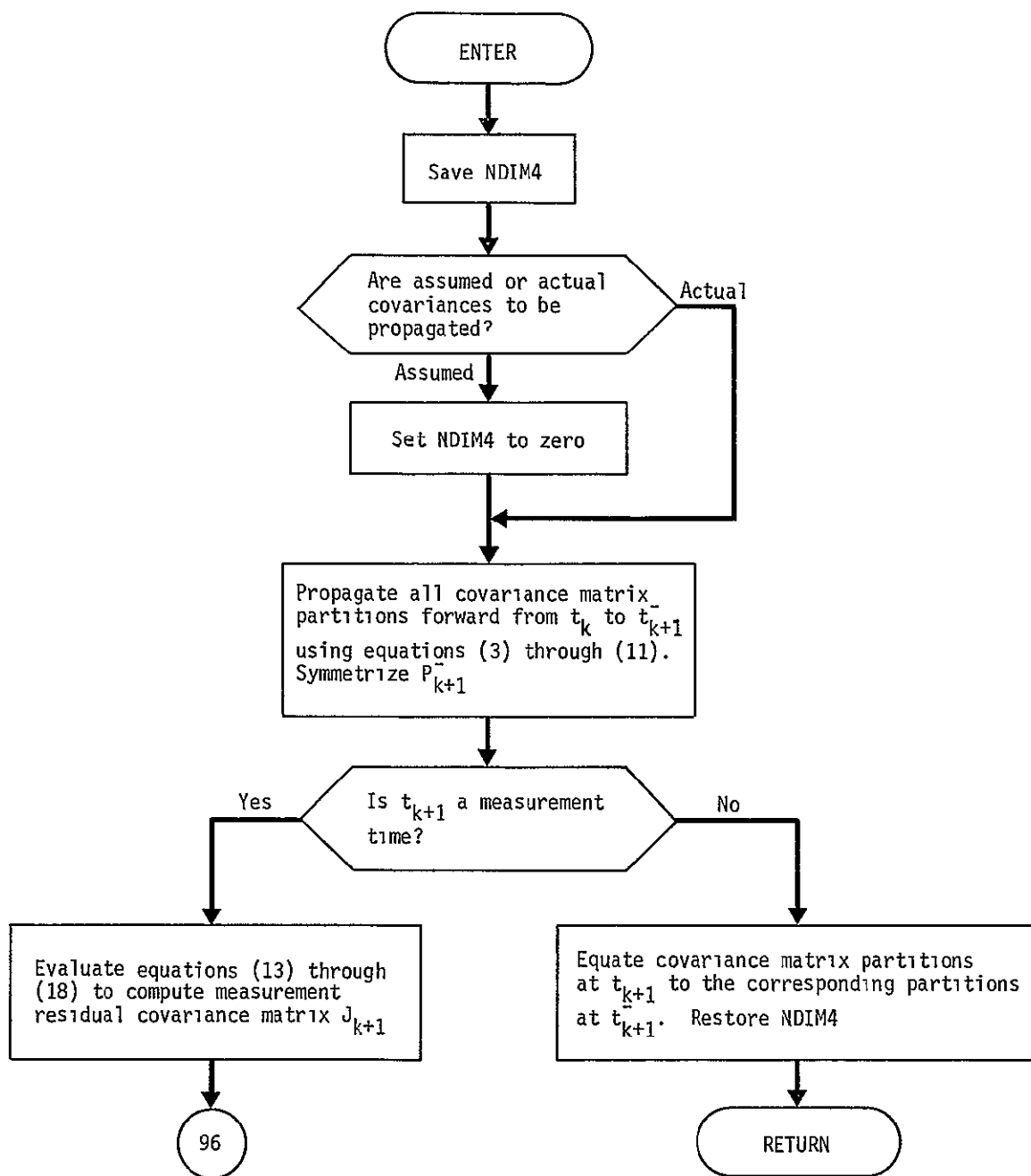
$$P_{s_{k+1}}^+ = P_{s_{k+1}}^- - S_{k+1} B^T - B S_{k+1}^T + S_{k+1} J_{k+1} S_{k+1}^T \quad (24)$$

$$C_{x_s u_{k+1}}^+ = C_{x_s u_{k+1}}^- - S_{k+1} D^T \quad (25)$$

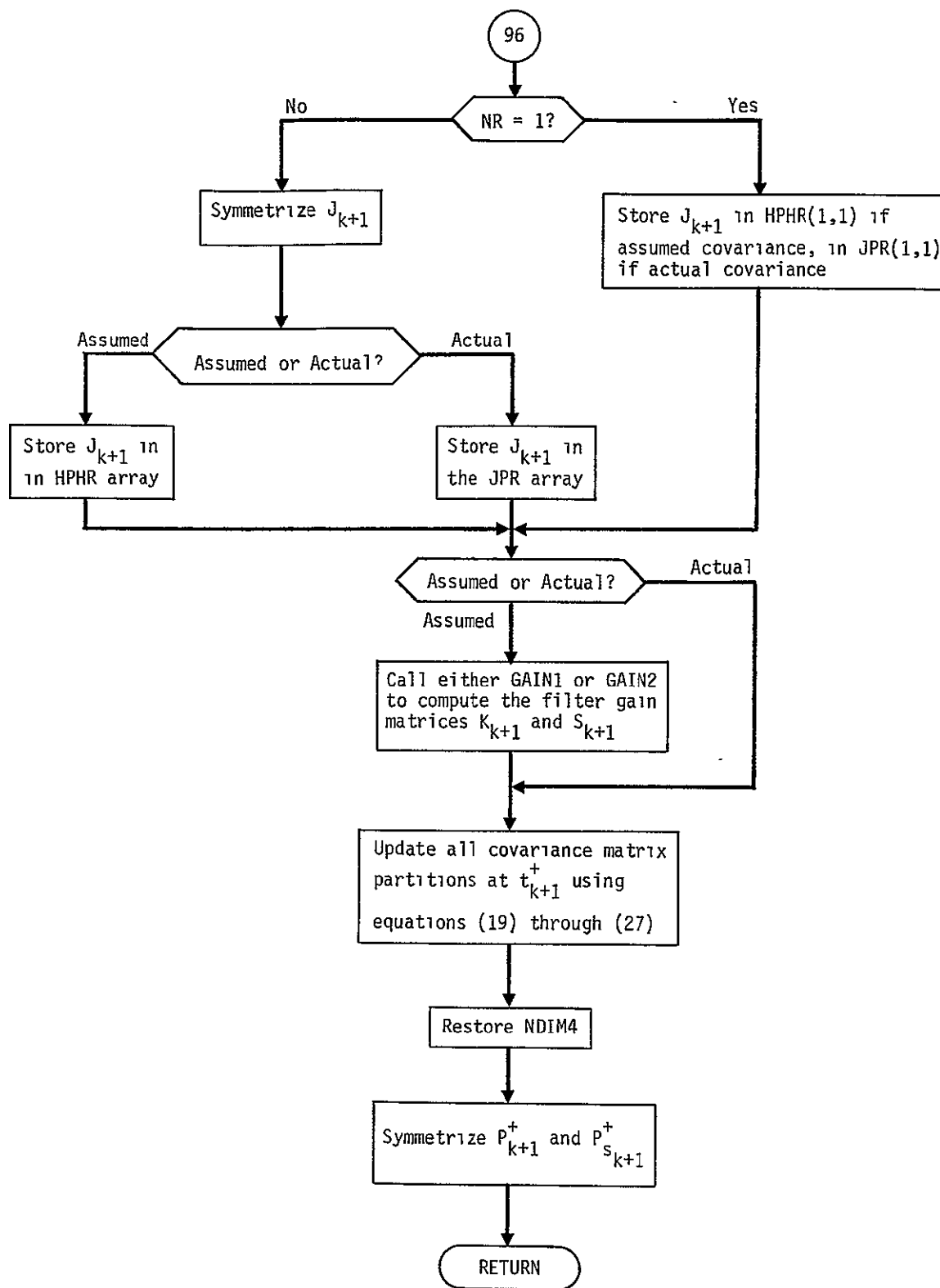
$$C_{x_s v}^{+}_{k+1} = C_{x_s v}^{-}_{k+1} - S_{k+1} E^T \quad (26)$$

$$C_{x_s w}^{+}_{k+1} = C_{x_s w}^{-}_{k+1} - S_{k+1} F^T \quad (27)$$

It should be noted that propagation equations (3) through (11) are also used to propagate both assumed control covariance and actual 2nd moment matrix partitions over the time interval separating two successive guidance events. The update equations, of course, are not used in this situation.



GNAVM Flow Chart



SUBROUTINE GPRINT

PURPOSE: TO PRINT ACTUAL ESTIMATION-ERROR STATISTICS

CALLING SEQUENCE: CALL GPRINT(IFLAG,TIMM)

ARGUMENTS: IFLAG I =3 PRINT ACTUAL STATISTICS AT A
 GUIDANCE EVENT
 =10 PRINT ACTUAL ESTIMATION ERROR
 STATISTICS
 =2 PRINT ACTUAL ESTIMATION ERROR
 STATISTICS AT A PREDICTION EVENT

TIMM I TIME TO BE PRINTED

SUBROUTINES SUPPORTED: PRED ERRANN SETEVN

SUBROUTINES REQUIRED: MOMENT

LOCAL SYMBOLS: A HOLLERITH WORD -AFTER-

B HOLLERITH WORD -BEFORE-

DUM INTERMEDIATE VECTOR

EXSTSV TEMPORARY STORAGE FOR EXST

EXTSV TEMPORARY STORAGE FOR EXT

ROW INTERMEDIATE VECTOR

ZZ INTERMEDIATE VARIABLE

COMMON USED:

CXSUP	CXSVP	CXUP	CXVP	CXXSP
EMRES	EU	EV	EW	EXI
EXSI	EXST	EXSTP	EXTP	GCXSU
GCXSV	GCXSW	GCXSWP	GCXU	GCXV
GCXW	GCXWP	GCXXS	GP	GPS
GU	GV	GW	JPR	NDIM1
NDIM2	NDIM3	NDIM4	NR	PP
PSP	RPR	TRTM2	XIG	XLAB
XSL	XU	XV		

SUBROUTINE GQCOMP

PURPOSE: TO COMPUTE ACTUAL EXECUTION ERROR STATISTICS

CALLING SEQUENCE: CALL GQCOMP(VV,EE,EEE,EV,Q)

ARGUMENTS: VV I ACTUAL COMMANDED VELOCITY CORRECTION
EE I MEANS OF ACTUAL EXECUTION ERRORS
EEE I 2ND MOMENTS OF ACTUAL EXECUTION ERRORS
EV O EXPECTED VALUE OF ACTUAL EXECUTION ERROR
Q O ACTUAL EXECUTION ERROR 2ND MOMENT MATRIX

SUBROUTINES SUPPORTED: GENGID

LOCAL SYMBOLS: FACTR INTERMEDIATE VARIABLE
RHOP MAGNITUDE OF VW VECTOR
RHOP2 RHOP**2
V1 V(1)**2
V2 V(2)**2
V3 V(3)**2
V4 V1*V2*V3
XI INTERMEDIATE VARIABLE
XMUP INTERMEDIATE VARIABLE
ZETA INTERMEDIATE VARIABLE

GQCØMP Analysis

Subroutine GQCØMP computes the actual execution error mean and 2nd moment matrix for use in the generalized covariance analysis of a guidance event. The actual execution error $\delta\Delta V'_j$ is assumed to have the form

$$\delta\Delta V'_j = k' \Delta\hat{V}'_j + s' \frac{\Delta\hat{V}'_j}{|\Delta\hat{V}'_j|} + \delta\Delta V'_{\text{pointing}} \quad (1)$$

where k' denotes the actual proportionality error; s' , the actual resolution error; $\delta\Delta V'_{\text{pointing}}$, the actual pointing error; and $\Delta\hat{V}'_j$, the actual commanded velocity correction.

The means of the three ecliptic components of $\delta\Delta V'_j$ are given as:

$$E[\delta\Delta V'_x] = \left(\bar{k}' + \frac{\bar{s}'}{\rho'} \right) \Delta\hat{V}'_x + \frac{\rho' \Delta\hat{V}'_y \bar{\delta\alpha}' + \Delta\hat{V}'_x \Delta\hat{V}'_z \bar{\delta\beta}'}{\mu'} \quad (2)$$

$$E[\delta\Delta V'_y] = \left(\bar{k}' + \frac{\bar{s}'}{\rho'} \right) \Delta\hat{V}'_y + \frac{\Delta\hat{V}'_y \Delta\hat{V}'_z \bar{\delta\beta}' - \rho' \Delta\hat{V}'_x \bar{\delta\alpha}'}{\mu'} \quad (3)$$

$$E[\delta\Delta V'_z] = \left(\bar{k}' + \frac{\bar{s}'}{\rho'} \right) \Delta\hat{V}'_z - \mu' \bar{\delta\beta}' \quad (4)$$

where $\rho' = |\Delta\hat{V}'|$, $\mu' = [\Delta\hat{V}'_x^2 + \Delta\hat{V}'_y^2]^{\frac{1}{2}}$, and $\delta\alpha'$ and $\delta\beta'$ are the actual pointing angle errors, and both $E(\)$ and $(\bar{\ })$ indicate mean values.

The actual execution error 2nd moment matrix is defined by

$$\tilde{Q}'_j = E \left[\delta\Delta V'_j \delta\Delta V'^T_j \right] \quad (5)$$

the elements Q'_{ik} of matrix Q'_j are given as:

$$\begin{aligned} \tilde{Q}'_{11} = & \xi' \Delta\hat{V}'_x^2 + \frac{1}{\mu'^2} \left(\rho'^2 \Delta\hat{V}'_y^2 \bar{\delta\alpha}' \bar{\delta\alpha}' + \Delta\hat{V}'_x \Delta\hat{V}'_z \bar{\delta\beta}' \bar{\delta\beta}' + \right. \\ & \left. 2\rho' \Delta\hat{V}'_x \Delta\hat{V}'_y \Delta\hat{V}'_z \bar{\delta\alpha}' \bar{\delta\beta}' \right) + \frac{2\Delta\hat{V}'_x}{\mu'} \xi' \left(\rho' \Delta\hat{V}'_y \bar{\delta\alpha}' + \Delta\hat{V}'_x \Delta\hat{V}'_z \bar{\delta\beta}' \right) \end{aligned} \quad (5)$$

$$\begin{aligned} \tilde{Q}'_{22} = & \xi' \Delta \hat{V}_y'^2 + \frac{1}{\mu'^2} \left(\Delta \hat{V}_y'^2 \Delta \hat{V}_z'^2 \overline{\delta \beta'} \overline{\delta \beta'} + \rho'^2 \Delta \hat{V}_x'^2 \overline{\delta \alpha'} \overline{\delta \alpha'} - \right. \\ & \left. 2\rho' \Delta \hat{V}_x' \Delta \hat{V}_y' \Delta \hat{V}_z' \overline{\delta \alpha'} \overline{\delta \beta'} \right) + \frac{2\Delta \hat{V}_y'}{\mu'} \zeta' \left(\Delta \hat{V}_y' \Delta \hat{V}_z' \overline{\delta \beta'} - \rho' \Delta \hat{V}_x' \overline{\delta \alpha'} \right) \quad (6) \end{aligned}$$

$$\tilde{Q}'_{33} = \xi' \Delta \hat{V}_z'^2 + \mu'^2 \overline{\delta \beta'} \overline{\delta \beta'} - 2\Delta \hat{V}_z' \mu' \zeta' \overline{\delta \beta'} \quad (7)$$

$$\begin{aligned} \tilde{Q}'_{12} = \tilde{Q}'_{21} = & \xi' \Delta \hat{V}_x' \Delta \hat{V}_y' + \frac{\zeta'}{\mu'} \left[2\Delta \hat{V}_x' \Delta \hat{V}_y' \Delta \hat{V}_z' \overline{\delta \beta'} - \rho' \left(\Delta \hat{V}_x'^2 - \Delta \hat{V}_y'^2 \right) \overline{\delta \alpha'} \right] + \\ & \frac{1}{\mu'^2} \left[-\rho'^2 \Delta \hat{V}_x' \Delta \hat{V}_y' \overline{\delta \alpha'} \overline{\delta \alpha'} + \rho' \Delta \hat{V}_z' \left(\Delta \hat{V}_y'^2 - \Delta \hat{V}_x'^2 \right) \overline{\delta \alpha'} \overline{\delta \beta'} + \right. \\ & \left. \Delta \hat{V}_x' \Delta \hat{V}_y' \Delta \hat{V}_z'^2 \overline{\delta \beta'} \overline{\delta \beta'} \right] \quad (8) \end{aligned}$$

$$\begin{aligned} \tilde{Q}'_{13} = \tilde{Q}'_{31} = & \xi' \Delta \hat{V}_x' \Delta \hat{V}_z' + \zeta' \left[\frac{\Delta \hat{V}_z'}{\mu'} \left(\rho' \Delta \hat{V}_y' \overline{\delta \alpha'} + \Delta \hat{V}_x' \Delta \hat{V}_z' \overline{\delta \beta'} \right) - \mu' \Delta \hat{V}_x' \overline{\delta \beta'} \right] \\ & - \rho' \Delta \hat{V}_y' \overline{\delta \alpha'} \overline{\delta \beta'} - \Delta \hat{V}_x' \Delta \hat{V}_z' \overline{\delta \beta'} \overline{\delta \beta'} \quad (9) \end{aligned}$$

$$\begin{aligned} \tilde{Q}'_{23} = \tilde{Q}'_{32} = & \xi' \Delta \hat{V}_y' \Delta \hat{V}_z' + \zeta' \left[\frac{\Delta \hat{V}_z'}{\mu'} \left(\Delta \hat{V}_y' \Delta \hat{V}_z' \overline{\delta \beta'} - \rho' \Delta \hat{V}_x' \overline{\delta \alpha'} \right) - \mu' \Delta \hat{V}_y' \overline{\delta \beta'} \right] \\ & + \rho' \Delta \hat{V}_x' \overline{\delta \alpha'} \overline{\delta \beta'} - \Delta \hat{V}_y' \Delta \hat{V}_z' \overline{\delta \beta'} \overline{\delta \beta'} \quad (10) \end{aligned}$$

where

$$\xi' = \overline{k'} \overline{k'} + \frac{2}{\rho'} \overline{k'} \overline{s'} + \frac{\overline{s'} \overline{s'}}{\rho'^2} \quad (11)$$

and

$$\zeta' = \overline{k'} + \frac{\overline{s'}}{\rho'} \quad (12)$$

SUBROUTINE GUID

PURPOSE COMPUTE GUIDANCE MATRIX, VARIATION MATRIX, AND TARGET
CONDITION COVARIANCE MATRIX AT A MIDCOURSE GUIDANCE
EVENT IN THE ERROR ANALYSIS PROGRAM

CALLING SEQUENCE: CALL GUID

SUBROUTINES SUPPORTED: GUIDM

SUBROUTINES REQUIRED: EPHEM ORB HYELS PARTL JACOBI PSIM MATIN STMPR NTM VARADA

LOCAL SYMBOLS	A	TWO-VARIABLE B-PLANE GUIDANCE SUB-MATRIX A
	BB	TWO-VARIABLE B-PLANE GUIDANCE SUB-MATRIX B
	BDRS	B DOT R
	BDR1	VALUE OF B DOT R RETURNED FROM PARTL (NOT USED)
	BDTS	B DOT T
	BDT1	VALUE OF B DOT T RETURNED FROM PARTL (NOT USED)
	BS	MAGNITUDE OF B VECTOR
	B1	VALUE OF B RETURNED FROM PARTL (NOT USED)
	D	INTERMEDIATE JULIAN DATE
	DUM1	ARRAY OF EIGENVECTORS
	EGVCT	ARRAY OF EIGENVECTORS
	EGVL	ARRAY OF EIGENVALUES
	ICS	INTERMEDIATE STORAGE FOR ICL2
	ICLS	INTERMEDIATE STORAGE FOR ICL
	INCMTS	INTERMEDIATE STORAGE FOR INCHT
	IPR	INTERMEDIATE STORAGE FOR IPRINT
	ISP	INTERMEDIATE STORAGE FOR ISP2
	PBR	PARTIAL OF B DOT R WITH RESPECT TO STATE VECTOR
	PBT	PARTIAL OF B DOT T WITH RESPECT TO STATE

VECTOR

PHI1	INTERMEDIATE ARRAY
PHI2	INTERMEDIATE ARRAY
PHI3	INTERMEDIATE ARRAY
RI	NOMINAL SPACECRAFT STATE AT GUIDANCE EVENT
ROW	TARGET CONDITION CORRELATION MATRIX
RTPS	INERTIAL SPACECRAFT STATE AT SPHERE OF INFLUENCE
SQP	TARGET CONDITION STANDARD DEVIATIONS
TCA	TRAJECTORY TIME AT CLOSEST APPROACH
TSI	TRAJECTORY TIME AT SPHERE OF INFLUENCE
XCA	INERTIAL SPACECRAFT STATE AT CLOSEST APPROACH
XSIP	SPACECRAFT POSITION RELATIVE TO TARGET PLANET AT SPHERE OF INFLUENCE
XSIV	SPACECRAFT VELOCITY RELATIVE TO TARGET PLANET AT SPHERE OF INFLUENCE

COMMON COMPUTED/USED:	ICL2	IPRINT	ISPH	ISP2	NO
	XP				
COMMON COMPUTED:	DELTH	EM	TRTM1	TSOI1	
COMMON USED:	ALNGTH	BDR	BDT	B	DATEJ
	DC	DSI	FNTM	FOV	F
	IBARY	ICL	IHYP1	ISTMC	NBOD
	NB	NTMC	NTP	ONE	PHI
	P	RC	RSI	TM	VSI
	ZERO				

GUID Analysis

Subroutine GUID is used in the error analysis mode to compute the same quantities which subroutine GUI5 computes in the simulation mode. Subroutine GUID differs from GUI5 in that instead of calling NTMS and VARSIM as does GUI5, subroutine GUID calls NTM and VARADA. In addition, the state transition and variation matrices computed in GUID are referenced to the targeted nominal since the most recent nominal is not defined for the error analysis mode. These differences entail only minor logic differences in the flow chart for GUID, and for this reason no GUID flow chart is presented. See subroutine GUI5 analysis and flow chart for further details.

SUBROUTINE GUIDM

PURPOSE CONTROL EXECUTION OF A GUIDANCE EVENT IN THE ERROR ANALYSIS PROGRAM

CALLING SEQUENCE: CALL GUIDM

SUBROUTINES SUPPORTED: ERRANN

SUBROUTINES REQUIRED:	CORREL	DYNO	GUID	HYELS	JACOBI
	NAVM	PSIM	STMPR		

LOCAL SYMBOLS:	ADA	VARIATION MATRIX
	AMAX	INTERMEDIATE VARIABLE USED TO FIND MAXIMUM EIGENVALUE OF VELOCITY CORRECTION COVARIANCE MATRIX (S MATRIX)
	CXSU1	STORAGE FOR CXSU KNOWLEDGE COVARIANCE
	CXSV1	STORAGE FOR CXSV KNOWLEDGE COVARIANCE
	CXU1	STORAGE FOR CXU KNOWLEDGE COVARIANCE
	CXV1	STORAGE FOR CXV KNOWLEDGE COVARIANCE
	CXXS1	STORAGE FOR CXXS KNOWLEDGE COVARIANCE
	DUM1	INTERMEDIATE VARIABLE
	DUM	VECTOR SUM OF UPDATE AND STATISTICAL VELOCITY CORRECTIONS
	EGM	MAXIMUM EIGENVALUE OF S MATRIX
	EGVCT	ARRAY OF EIGENVECTORS
	EGVL	ARRAY OF EIGENVALUES
	EXEC	EXECUTION ERROR COVARIANCE MATRIX
	EXV	EXPECTED VALUE OF VELOCITY CORRECTION
	GA	GUIDANCE MATRIX
	GAP	INTERMEDIATE ARRAY EQUAL TO GA TIMES P
	ICODE	INTERNAL CONTROL FLAG
	ICODE2	INTERNAL CONTROL FLAG
	IGP	MIDCOURSE GUIDANCE POLICY CODE

IQP	EXECUTION ERROR CODE
ISPHC	TEMPORARY STORAGE FOR ISPH
MAP	INDEX OF MAXIMUM EIGENVALUE OF S
OUT	SPACECRAFT VELOCITY RELATIVE TO TARGET PLANET IN PLANETO-CENTRIC EQUATORIAL COORDINATES
PS1	STORAGE FOR PS KNOWLEDGE COVARIANCE
P1	STORAGE FOR P KNOWLEDGE COVARIANCE
RF	NOMINAL TRAJECTORY STATE AT GUIDANCE EVENT
RHO	MAGNITUDE OF STATISTICAL DELTA-V
ROW	INTERMEDIATE VECTOR
SDV	STANDARD DEVIATION OF MAGNITUDE OF STATISTICAL DELTA-V
SQP	INTERMEDIATE VECTOR
TRS	TRACE OF S MATRIX
U	INTERMEDIATE VARIABLE
VEIG	MATRIX TO BE DIAGONALIZED
Z	INTERMEDIATE ARRAY

COMMON COMPUTED/USED:	CXSUG	CXSU	CXSVG	CXSV	CXUG
	CXU	CXVG	CXV	CXXSG	CXXS
	ISPH	NGE	PG	PSG	PS
	P	TG	XG		

COMMON COMPUTED:	DELTM	TRTM1	XI
------------------	-------	-------	----

COMMON USED:	FOP	FOV	ICDQ3	ICDT3	IEIG
	IHYP1	ISTMC	NDIM1	NDIM2	NDIM3
	ONE	Q	SIGALP	SIGBET	SIGPRO
	SIGRES	TWO	U0	V0	XF
	ZERO				

GUIDM Analysis

Subroutine GUIDM is the executive guidance subroutine in the error analysis program. In addition to controlling the computational flow for all types of guidance events, GUIDM also performs many of the required guidance computations itself.

Before considering each type of guidance event, the treatment of a general guidance event will be discussed. Let t_j be the time at which the guidance event occurs. Before any guidance event can be executed the targeted nominal state \bar{X}_j^- , knowledge covariance $P_{K_j}^-$, and control covariance $P_{C_j}^-$ must all be available, where $()^-$ indicates values immediately before the event. The first two quantities are available prior to entering GUIDM. However, GUIDM controls the propagation of the control covariance over the interval $[t_{j-1}, t_j]$, where t_{j-1} denotes the time of the previous guidance event.

The next step in the treatment of a general guidance event is concerned with the computation of the commanded velocity correction and the execution error covariance. In the error analysis program a non-statistical velocity correction is computed whenever the nominal target conditions are changed; otherwise, only a statistical velocity correction can be computed. The commanded velocity correction $\Delta \hat{V}_j$ is then used to compute the execution error covariance matrix \tilde{Q}_j . A summary of the execution error model and the equations used to compute \tilde{Q}_j can be found in the subroutine QC0MP' analysis section.

The last step is concerned with the updating of required quantities prior to returning to the basic cycle. An assumption underlying the modeled guidance process is that the targeted nominal is always updated by the commanded velocity correction. In the error analysis program only the non-statistical component is used to perform the state update and is indicated by the variable $\Delta \hat{V}_{UP_j}$. Thus, the targeted nominal state immediately following the guidance event is given by

$$\bar{X}_j^+ = \bar{X}_j^- + \begin{bmatrix} 0 \\ -\Delta \hat{V}_{UP_j} \end{bmatrix}.$$

The knowledge covariance is updated using the equation

$$P_{K_j}^+ = P_{K_j}^- + \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & \tilde{Q}_j \end{bmatrix}$$

if an impulsive thrust model is assumed. If the thrust is modeled as a series of impulses, then an effective execution error covariance \tilde{Q}_{eff} is computed and the knowledge covariance is updated using the equation

$$P_{K_j}^+ = P_{K_j}^- + \tilde{Q}_{eff}.$$

In either case the control covariance is updated simply by setting

$$P_{c_j}^+ = P_{K_j}^+.$$

This equation is a direct consequence of the assumption that the targeted nominal state is always updated at a guidance event.

A "compute only" option is available in GUIDM in which all of the $(\cdot)^+$ quantities will still be computed and printed. However, the state and all covariances are then reset to their former $(\cdot)^-$ values prior to returning to the basic cycle.

Each specific type of guidance event involves the computation of other quantities not discussed above. These will be covered in the following discussion of specific guidance events.

1. Midcourse and Biased Aimpoint Guidance

Linear midcourse guidance policies have form

$$\Delta \hat{V}_{N_j} = \Gamma_j \delta \hat{x}_j$$

where the subscript N indicates that this is the velocity correction required to null out deviations from the nominal target state. This notation is required to differentiate between this type of velocity correction and velocity corrections required to achieve an altered target

state. Linear midcourse guidance policies are discussed in more detail in the subroutine GUIIS analysis section.

Subroutine GUIDM calls GUID to compute the guidance matrix, Γ_j , and the target condition covariance immediately prior to the guidance event, W_j , and then uses Γ_j to compute the velocity correction covariance S_j , which is defined as

$$S_j = E \left[\begin{array}{cc} \hat{\Delta V}_{N_j} & \hat{\Delta V}_{N_j}^T \end{array} \right],$$

and is given by the equation

$$S_j = \Gamma_j (P_{c_j}^- - P_{K_j}^-) \Gamma_j^T.$$

This equation assumes that an optimal estimation algorithm is employed in the navigation process, since the derivation of this equation requires the orthogonality of the estimate and the estimation error.

In the error analysis program $\hat{\Delta V}_{N_j}$ is never available since no estimates $\hat{\delta X}_j$ are ever generated. Only the ensemble statistics of $\hat{\delta X}_j$ are available, which means only a statistical or effective velocity correction " $E[\hat{\Delta V}_{N_j}]$ " can be computed. In the STEAP error analysis program this effective velocity correction is assumed to have form

$$E[\hat{\Delta V}_{N_j}] = \rho_j \frac{\alpha_j}{|\alpha_j|}.$$

The magnitude ρ_j is given by the Hoffman-Young approximation

$$\rho_j = \sqrt{\frac{2A}{\pi}} \left(1 + \frac{B(\pi-2)}{A^2 \sqrt{5.4}} \right)$$

where

$$A = \text{trace } S_j = \lambda_1 + \lambda_2 + \lambda_3,$$

$$B = \lambda_1 \lambda_2 + \lambda_1 \lambda_3 + \lambda_2 \lambda_3,$$

and $\lambda_1, \lambda_2, \lambda_3$ are the eigenvalues of S_j . The direction of the effective velocity correction is assumed to coincide with the eigenvector corresponding to the maximum eigenvalue of S_j . This eigenvector is denoted by α_j . An alternate model assumes the direction coincides with the vector $(\lambda_1, \lambda_2, \lambda_3)$.

If planetary quarantine constraints must be satisfied at a midcourse correction, GUIDM calls BIAIM to compute the new aimpoint μ_j and the (non-statistical) bias velocity correction $\Delta \hat{V}_{B_j}$. All computations in BIAIM are based on linear guidance theory. However, an option is available in GUIDM to recompute $\Delta \hat{V}_{B_j}$, but not μ_j , using nonlinear techniques. This option is recommended if a biased aimpoint guidance event occurs at $t_j = \text{injection time}$. It should also be noted that \tilde{Q}_j is set to zero if $t_j = \text{injection time}$ since it is assumed that the injection covariance does not change for small changes in injection velocity.

After the updated control covariance P_c^+ has been computed, the target condition covariance matrix W_j^+ following the guidance correction is computed using the equation

$$(W_j^+)^T = \eta_j P_c^+ \eta_j^T$$

where variation matrix η_j has been previously computed in subroutine GUID.

2. Re-targeting

In the error analysis (and simulation) program a re-targeting event is defined to be the computation of a velocity correction $\Delta \hat{V}_{RT}$ required to achieve a new set of target conditions using nonlinear techniques. Since the original targeted nominal will be used as the zero-th iterate in the re-targeting process, the new target conditions must be close enough to the original nominal target condition to ensure a convergent process.

It should be noted that after a re-targeting event the new target conditions are henceforth treated as the nominal target conditions.

3. Orbital insertion

An orbital insertion event is divided into a decision event and an execution event. At a decision event the orbital insertion velocity correction $\Delta \hat{V}_{OI}$ and the time interval Δt separating decision and execution are computed based on the targeted nominal state at t_j . The relevant equations can be found in the subroutine CØPINS analysis section for coplanar orbital insertion; in NØPINS, for non-planar orbital insertion. Before returning to the basic cycle, GUIDM schedules the orbital insertion execution event to occur at $t_j + \Delta t$ and re-orders the necessary event arrays accordingly.

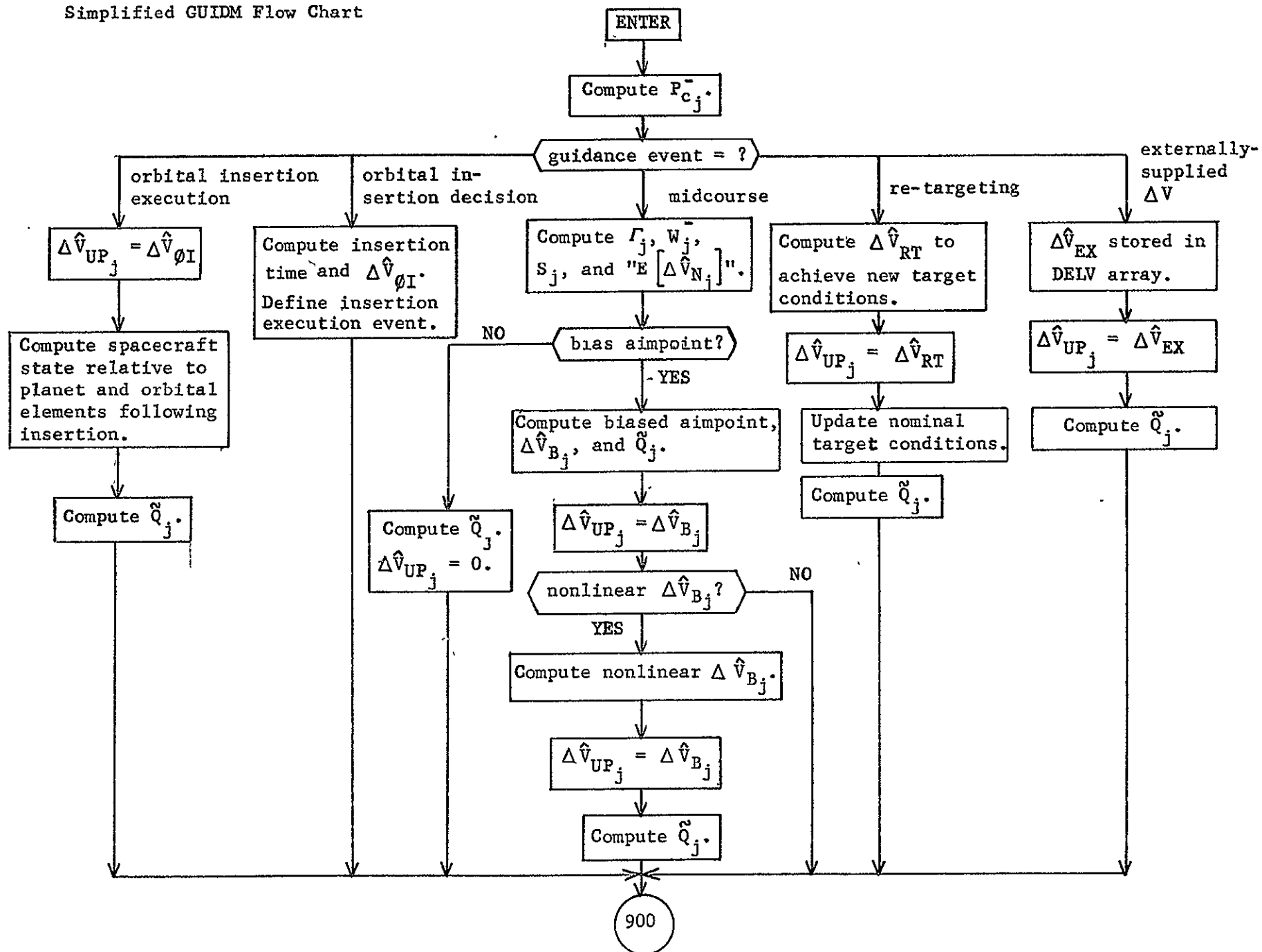
At an orbital insertion execution event the targeted nominal state is updated using the previously computed $\Delta \hat{V}_{OI}$. In addition, the planeto-centric equatorial components of $\Delta \hat{V}_{OI}$ and the nominal spacecraft cartesian and orbital element state following the insertion maneuver are computed.

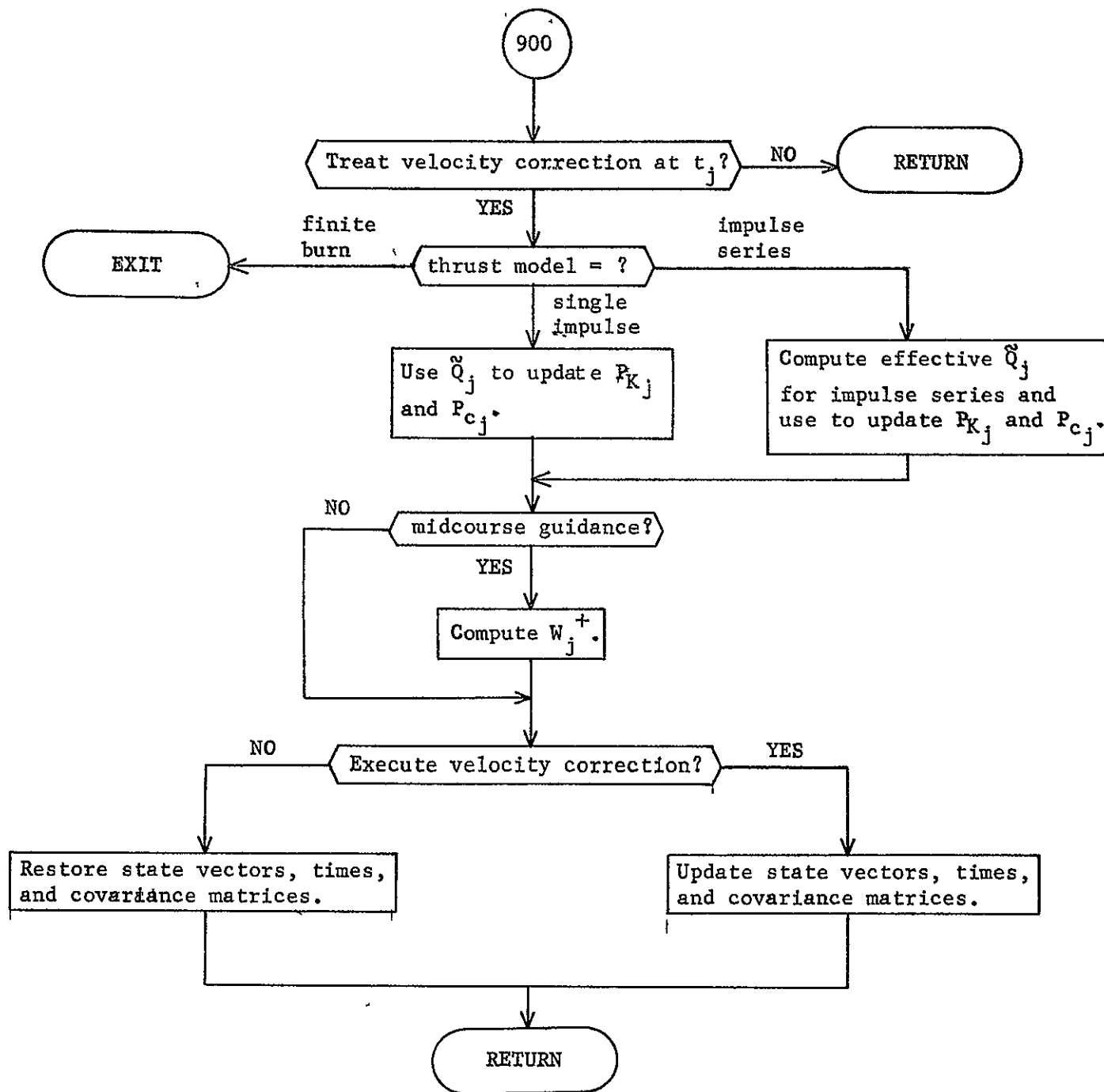
4. Externally-supplied velocity correction

At this type of guidance event the targeted nominal state is simply updated using the externally-supplied velocity correction $\Delta \hat{V}_{EX}$.

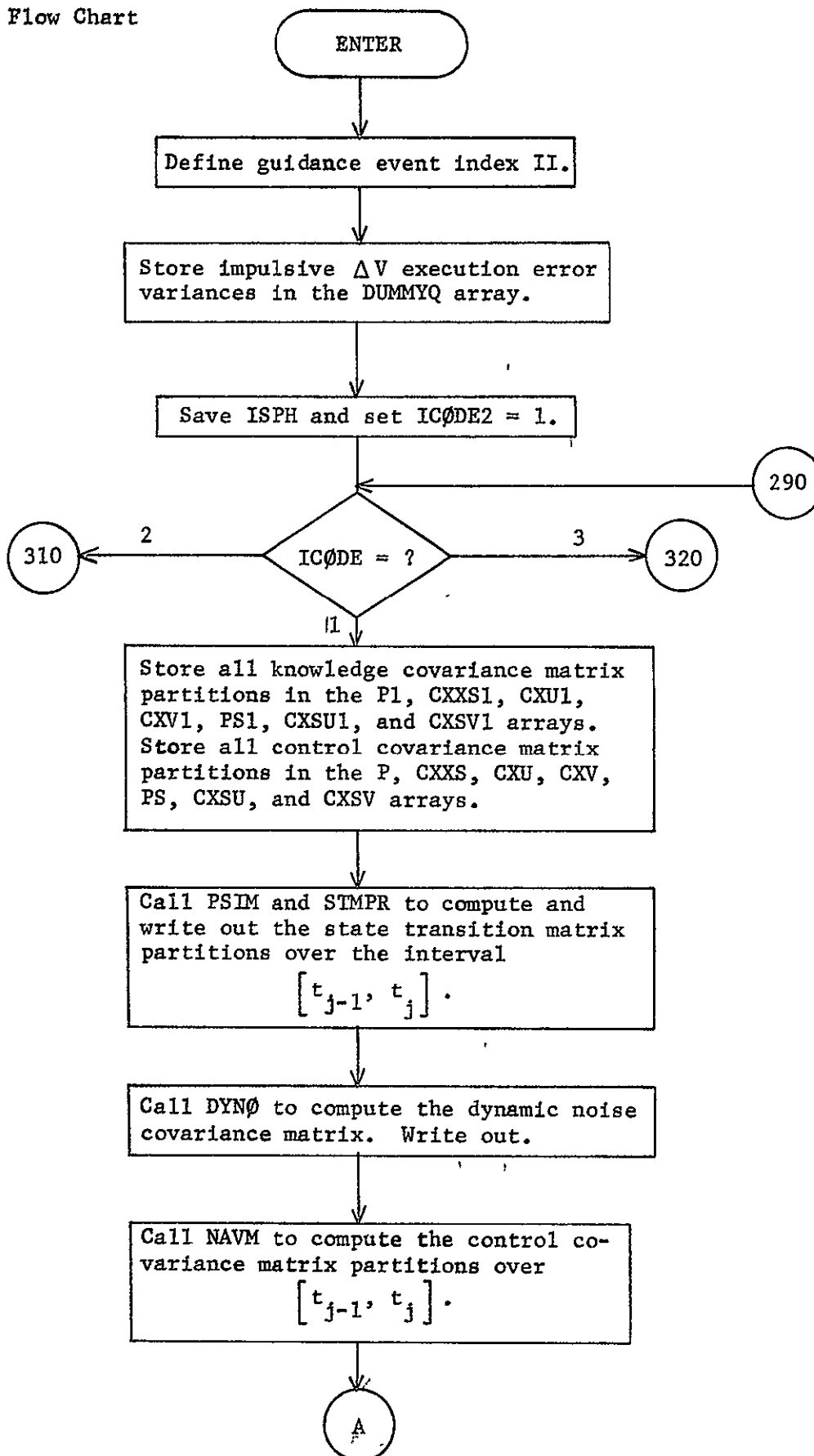
Because of the complexity of the GUIDM flow chart, a simplified flow chart depicting the main elements of the GUIDM structure precedes the complete GUIDM flow chart.

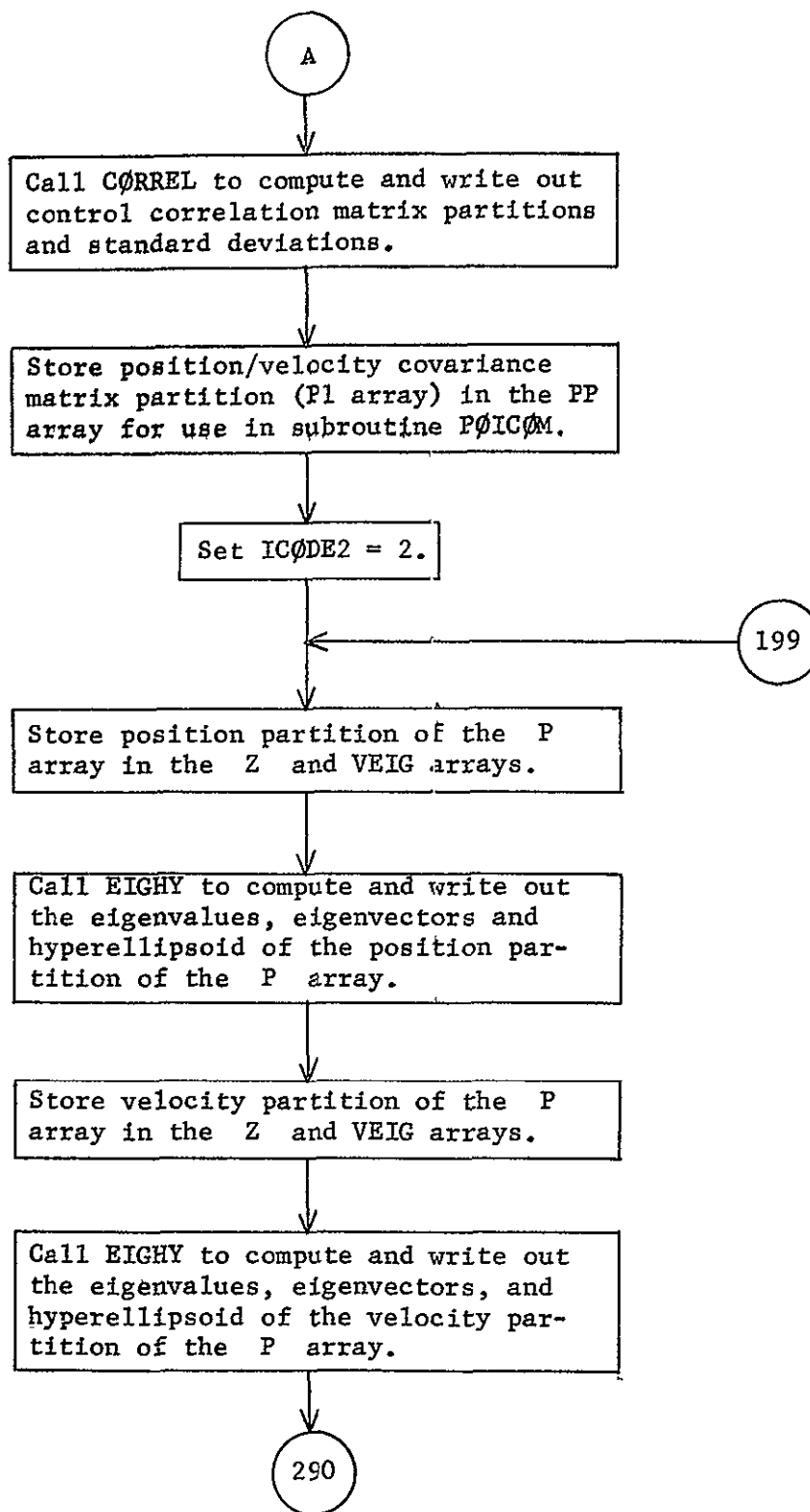
Simplified GUIDM Flow Chart

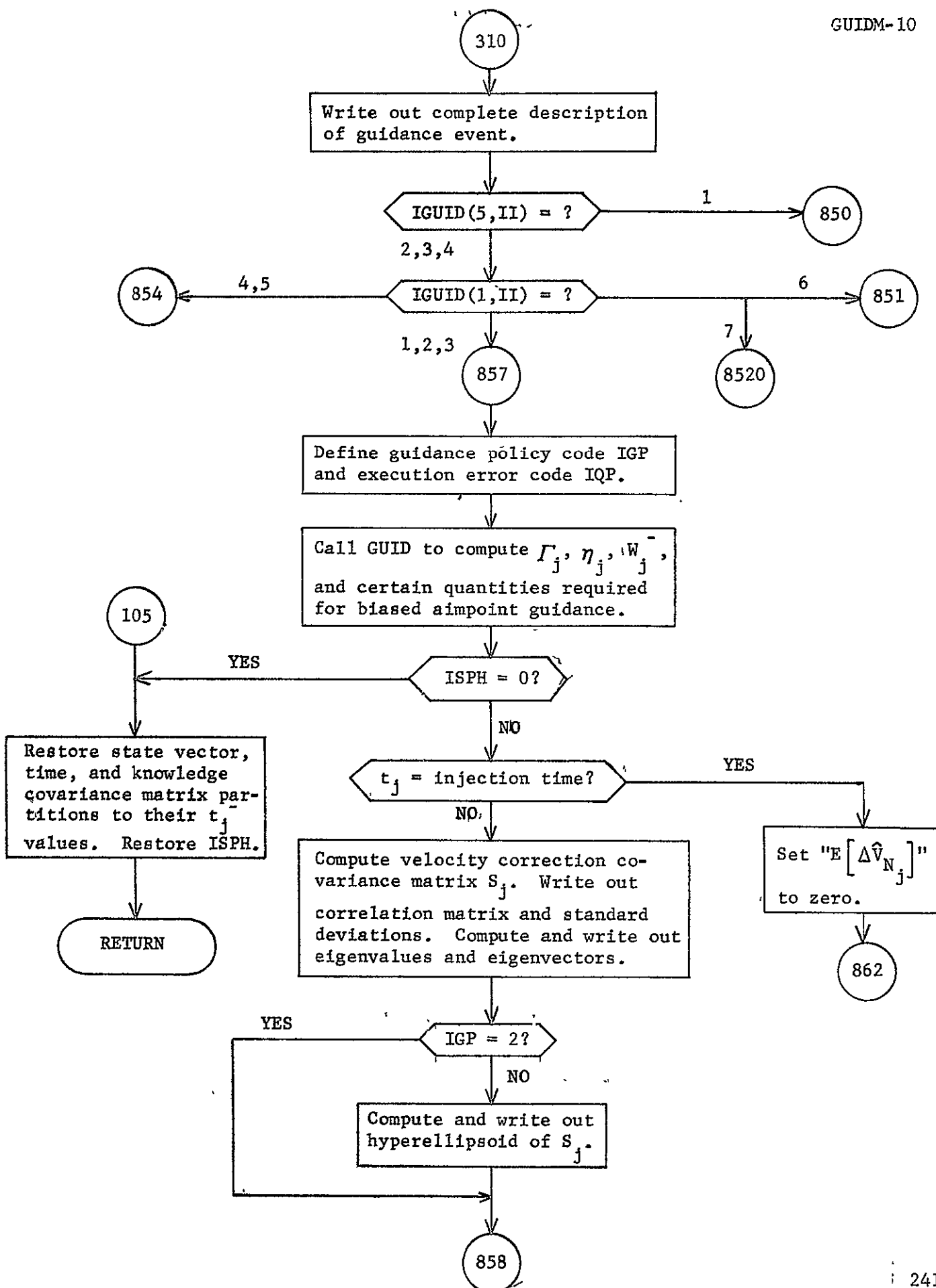


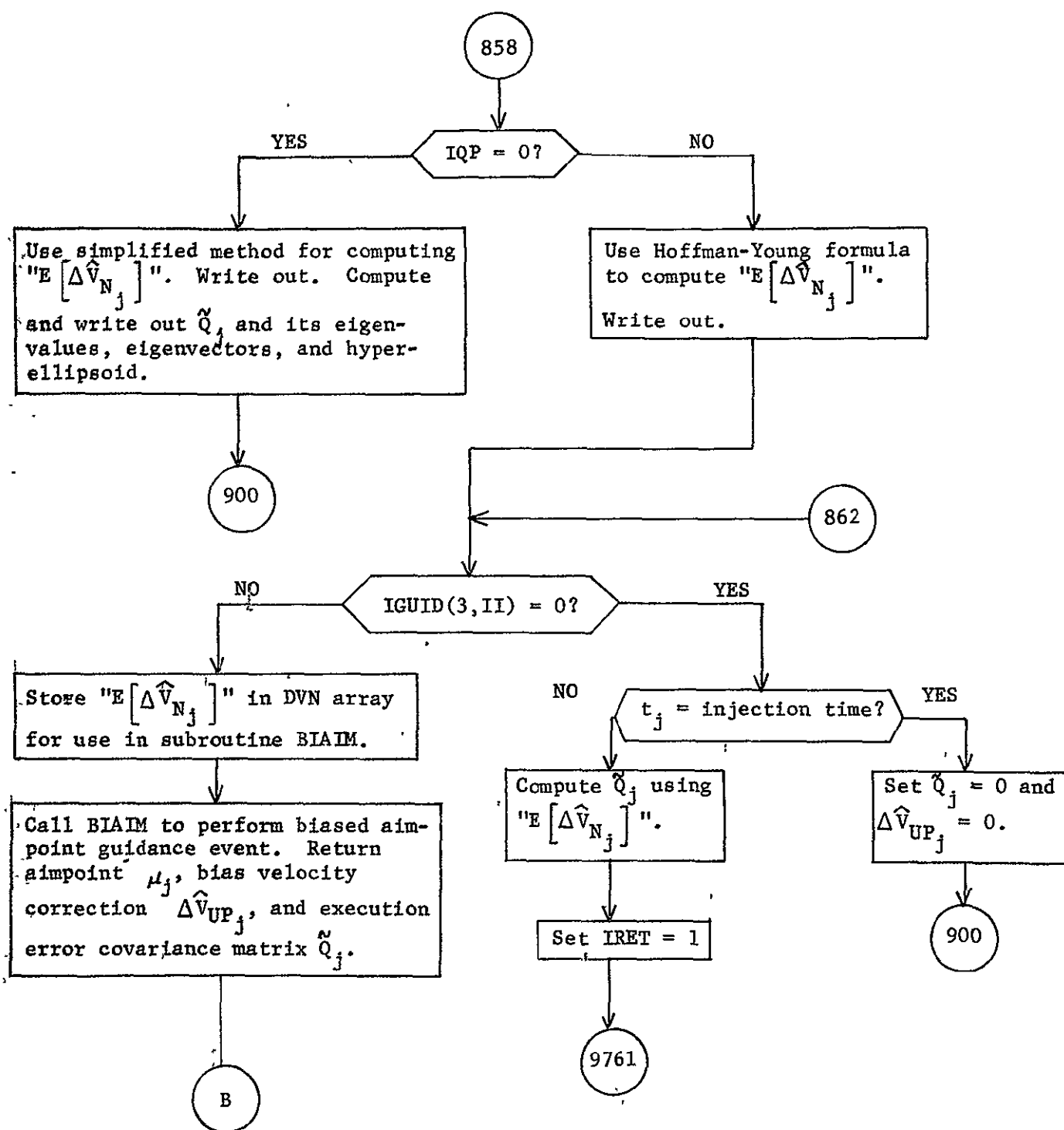


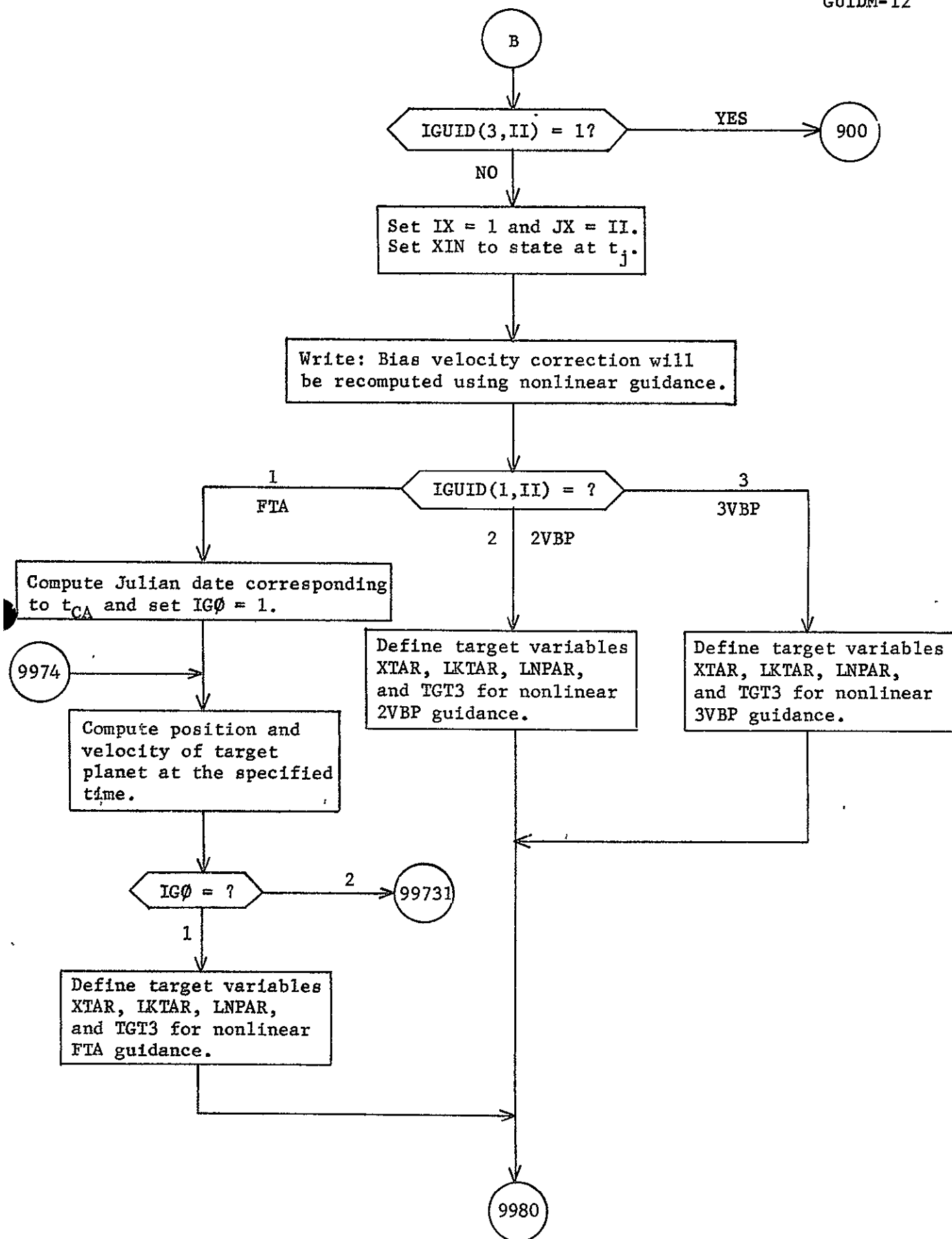
GUIDM Flow Chart

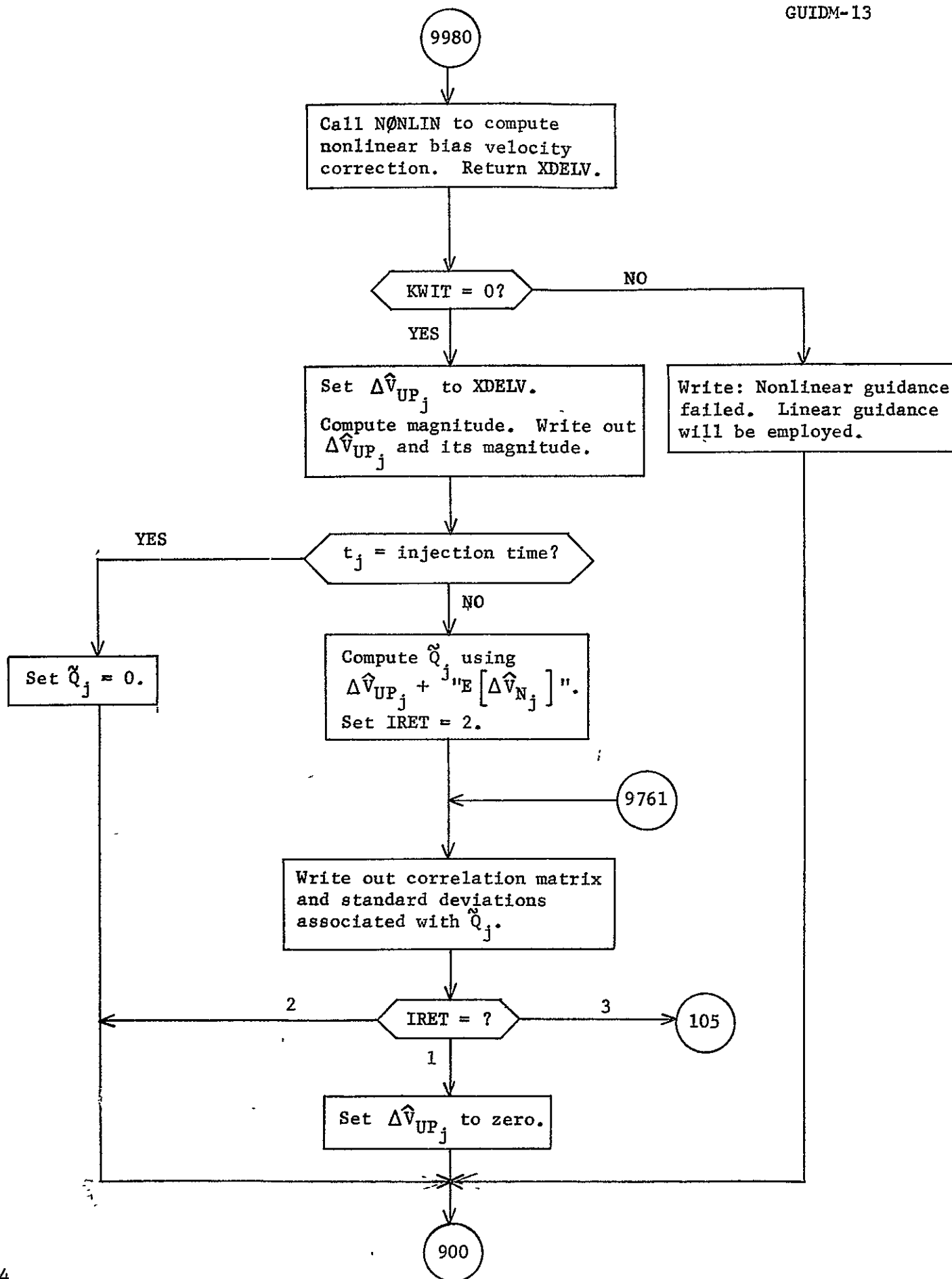


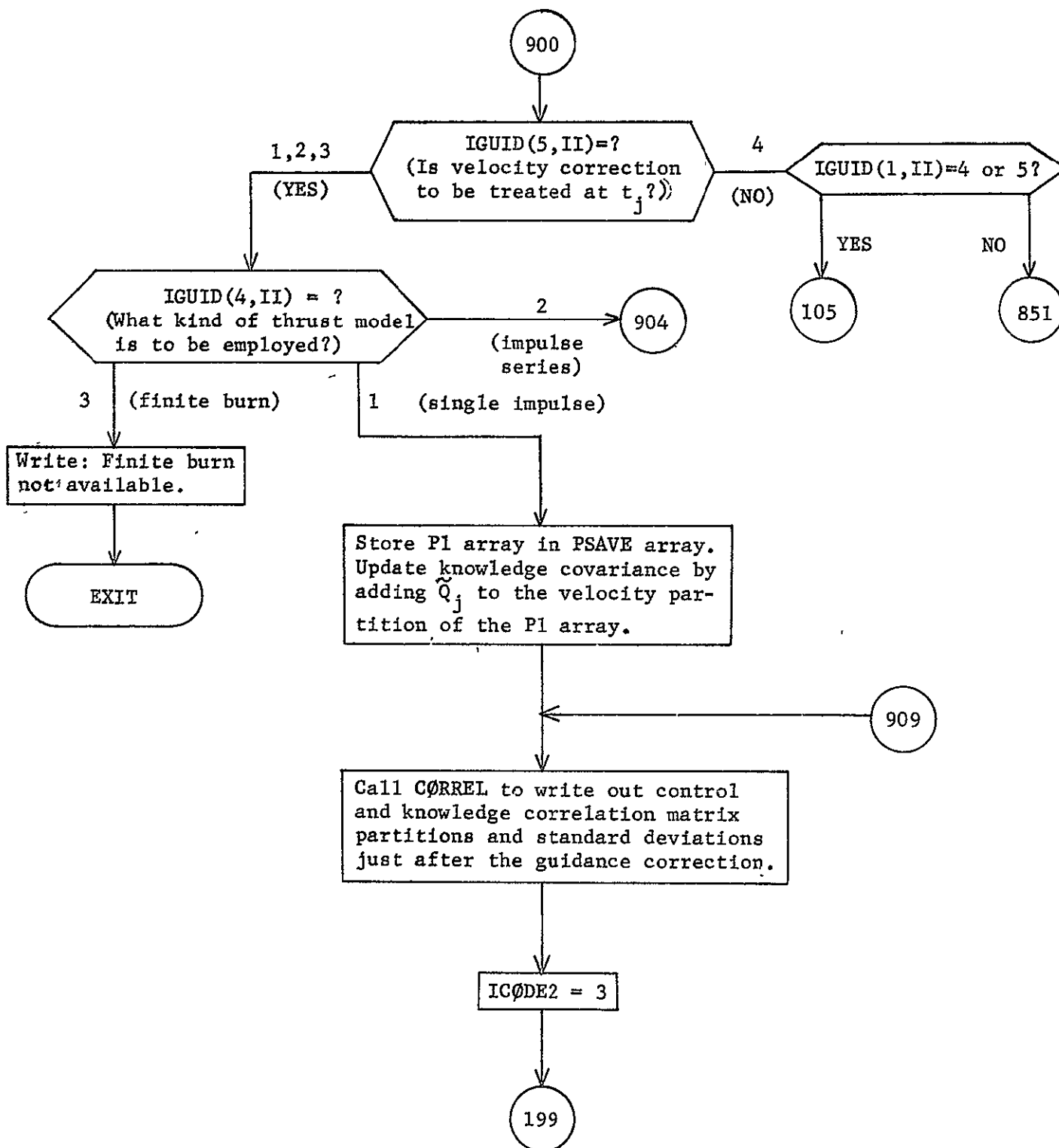


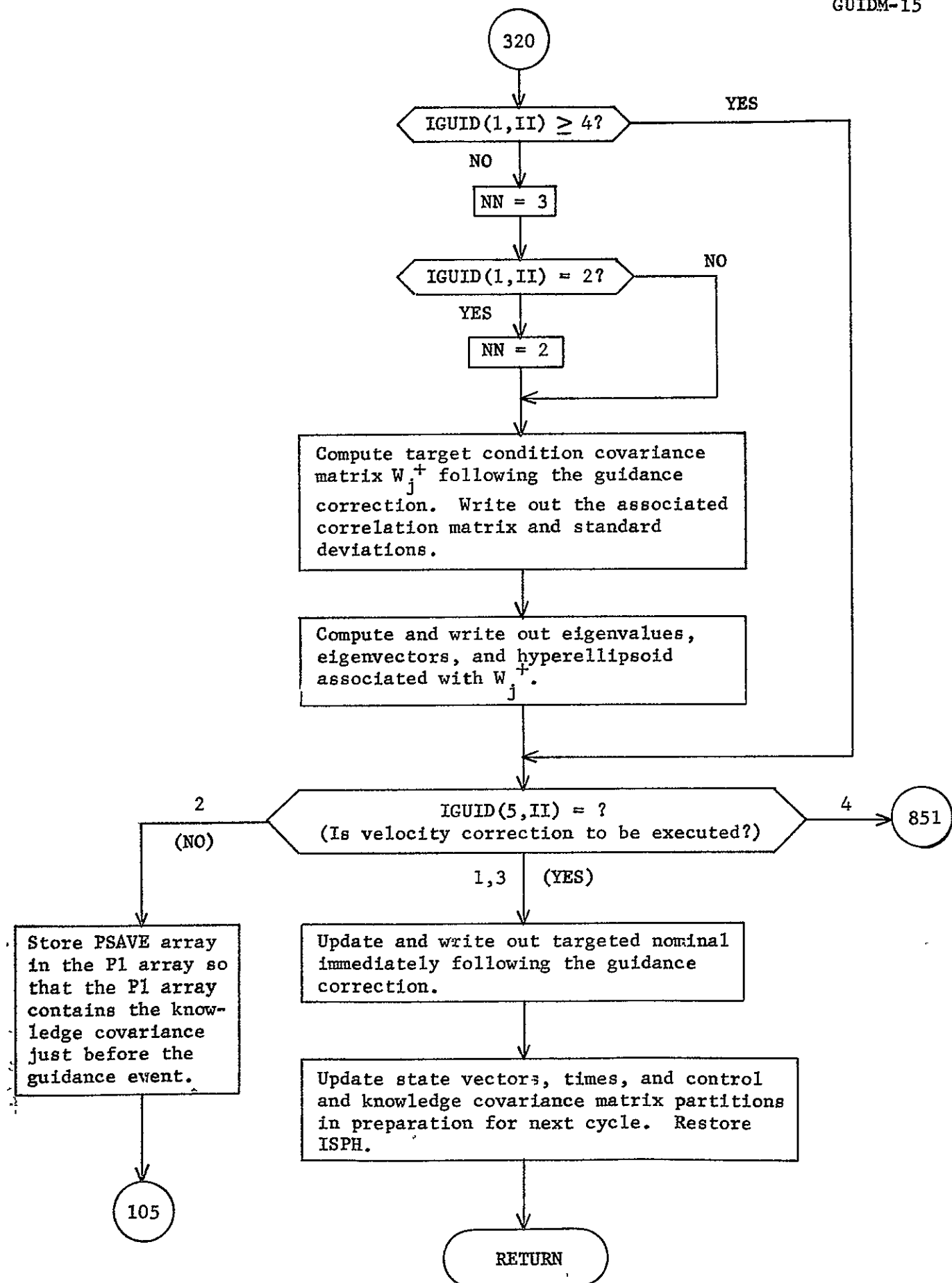


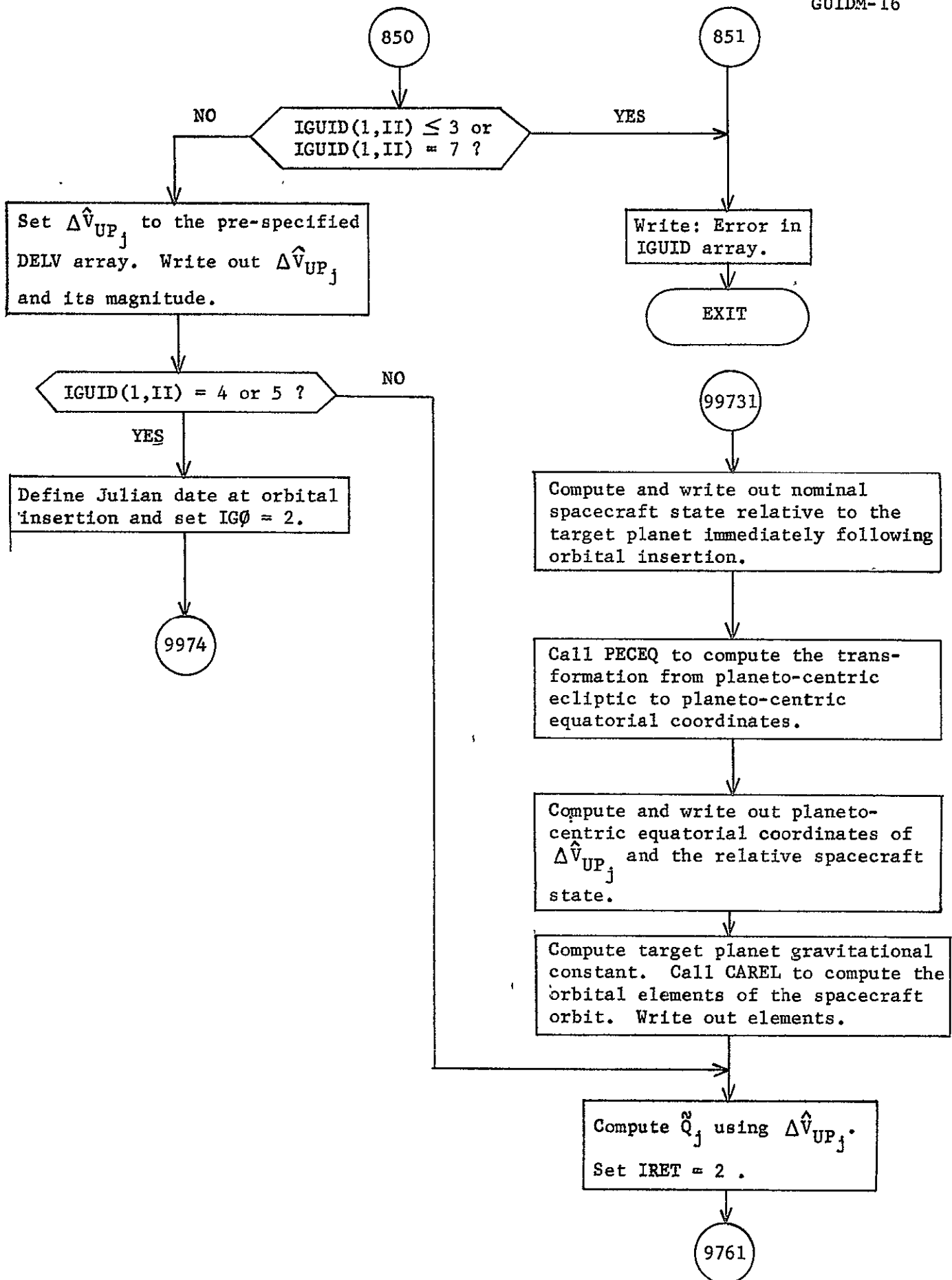


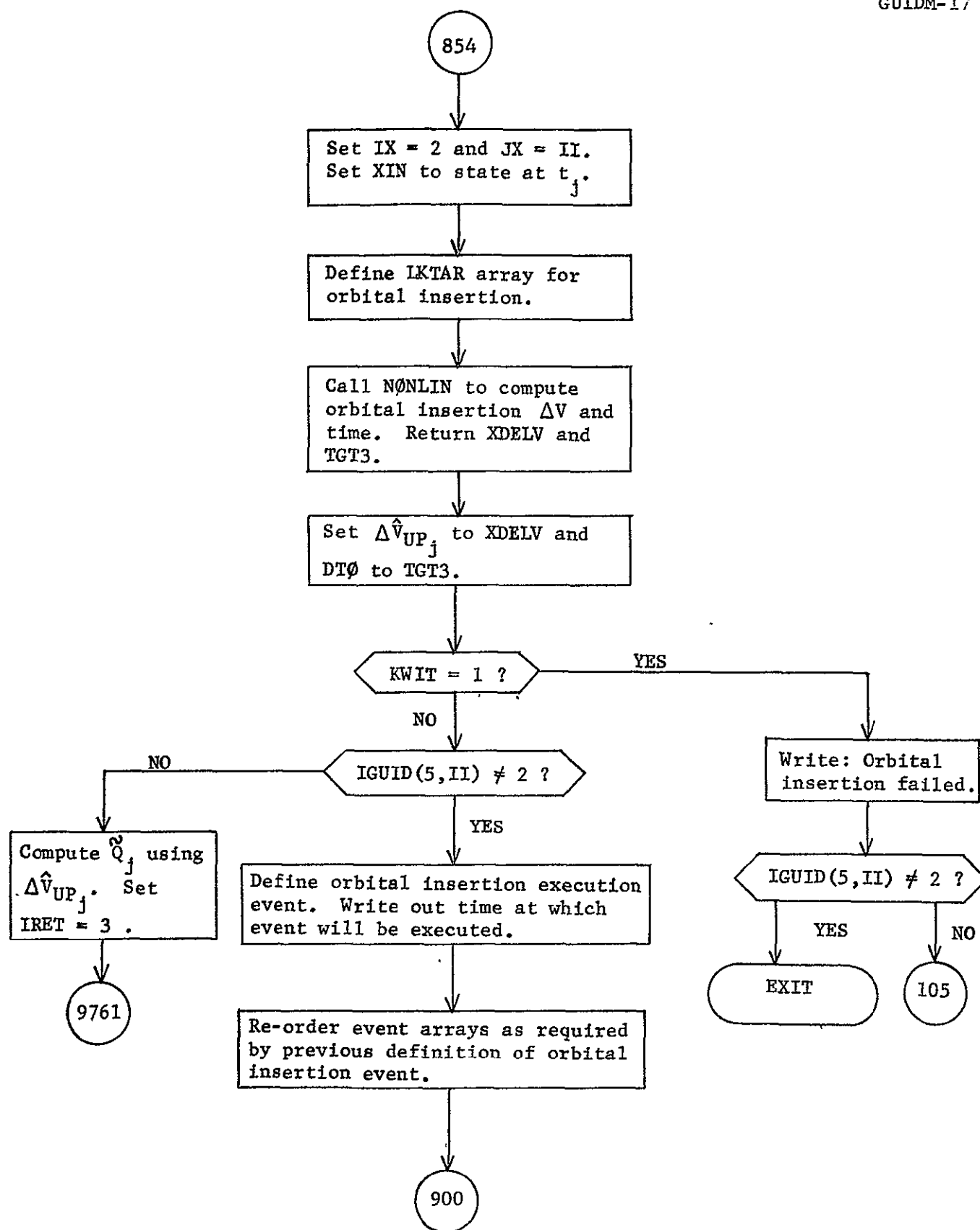


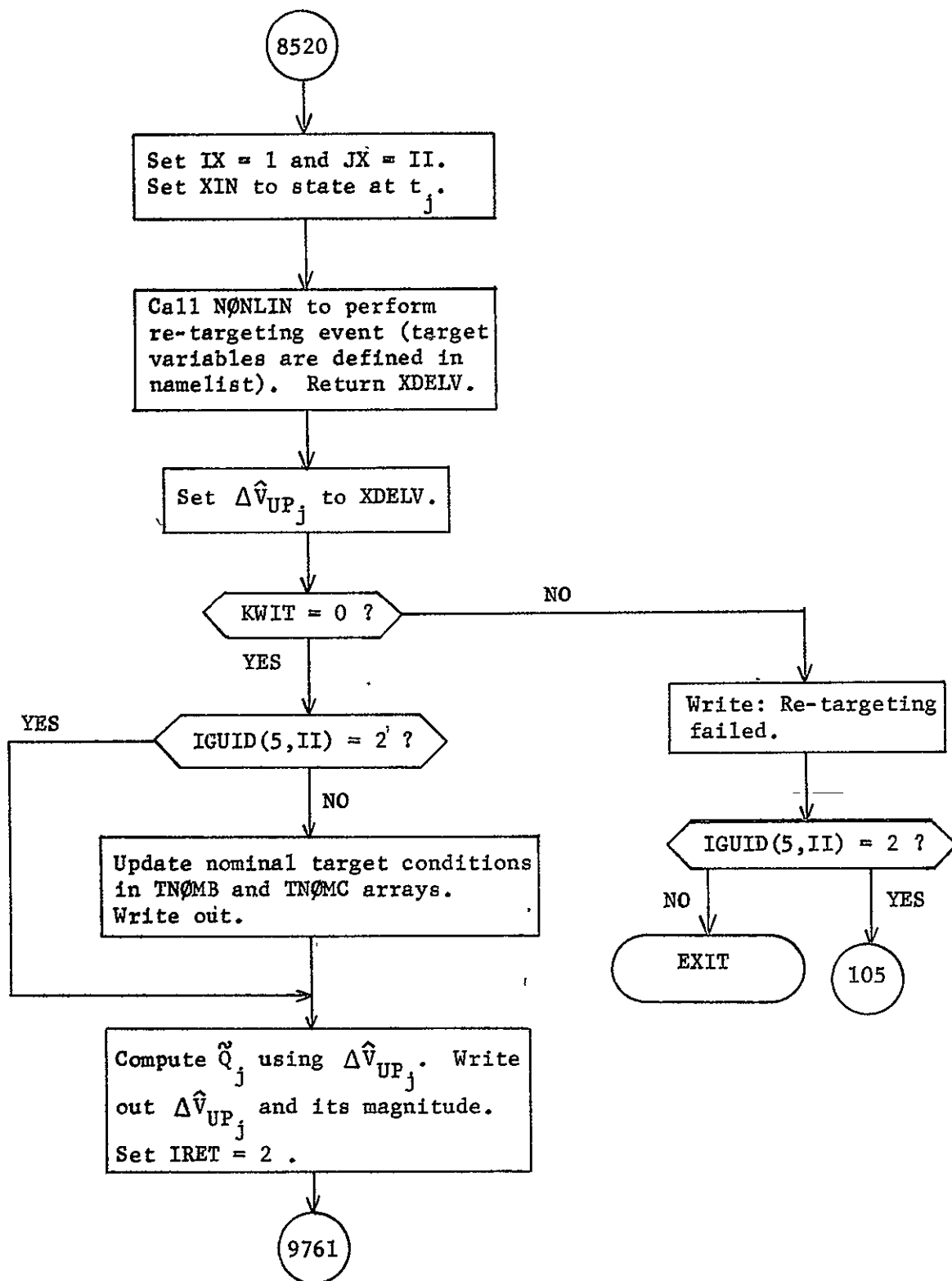


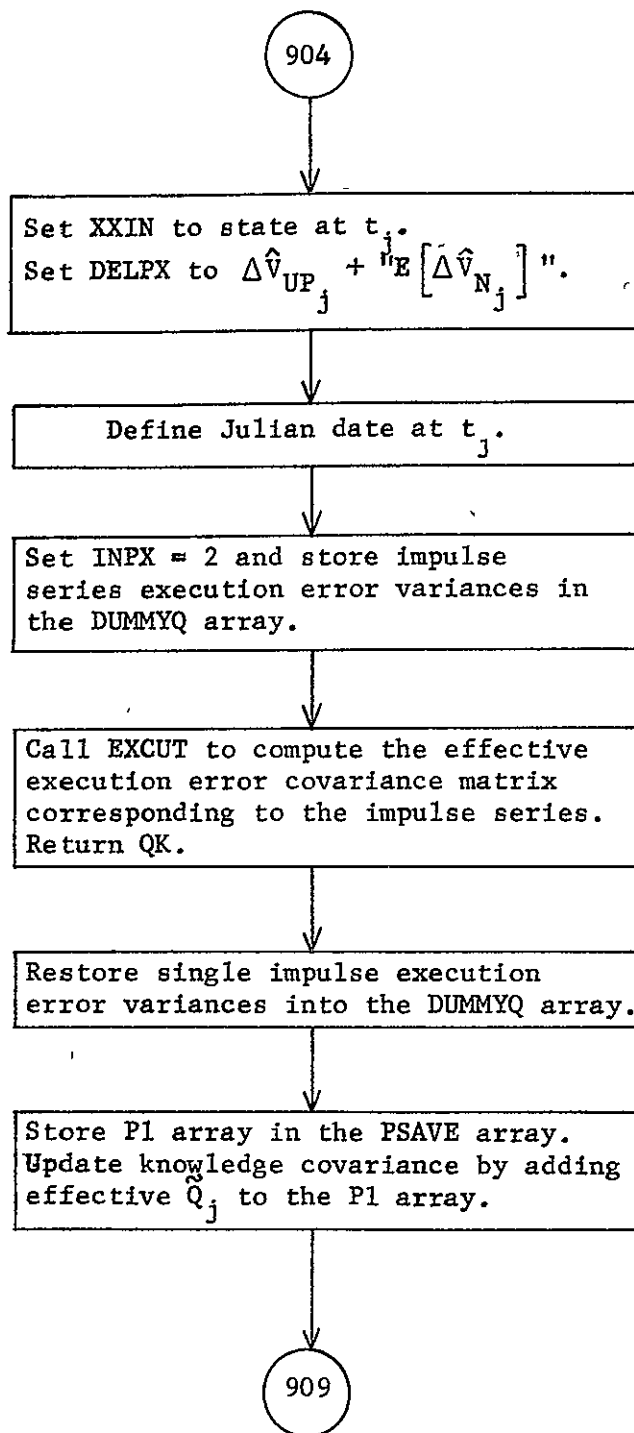












PROGRAM GUIS

**PURPOSE COMPUTE GUIDANCE MATRIX, VARIATION MATRIX, AND TARGET
 CONDITION COVARIANCE MATRIX AT A MIDCOURSE GUIDANCE
 EVENT IN THE SIMULATION PROGRAM**

SUBROUTINES SUPPORTED: GUISIM

**SUBROUTINES REQUIRED: EPHEM HYELS JACOBI MATIN NTMS
 ORB PARTL PSIM STMPR VARSIM**

LOCAL SYMBOLS A TWO-VARIABLE B-PLANE GUIDANCE SUB-MATRIX A

BB TWO-VARIABLE B-PLANE GUIDANCE SUB-MATRIX B

**BDR1 VALUE OF B DOT R RETURNED FROM PARTL (NOT
 USED)**

**BDT1 VALUE OF B DOT T RETURNED FROM PARTL (NOT
 USED)**

**B1 MAGNITUDE OF B VECTOR RETURNED FROM PARTL
 (NOT USED)**

DUM1 ARRAY OF EIGENVECTORS

DUM INTERMEDIATE ARRAY

EGVCT ARRAY OF EIGENVECTORS

EGVL ARRAY OF EIGENVALUES

ICLS INTERMEDIATE STORAGE FOR ICL

ICS INTERMEDIATE STORAGE FOR ICL2

IPR INTERMEDIATE STORAGE FOR IPRINT

ISPS INTERMEDIATE STORAGE FOR ISP2

**PBR' PARTIAL OF B DOT R WITH RESPECT TO STATE
 VECTOR**

**PBT PARTIAL OF B DOT I WITH RESPECT TO STATE
 VECTOR**

PHI1 INTERMEDIATE ARRAY

PHI2 INTERMEDIATE ARRAY

PHI3 INTERMEDIATE ARRAY

RI1 MOST RECENT NOMINAL SPACECRAFT STATE AT
GUIDANCE EVENT

RI TARGETED NOMINAL SPACECRAFT STATE AT
GUIDANCE EVENT

RMCA SPACECRAFT DISTANCE FROM TARGET PLANET AT
CLOSEST APPROACH

RMSI SPACECRAFT DISTANCE FROM TARGET PLANET AT
SPHERE OF INFLUENCE

ROW TARGET CONDITION CORRELATION MATRIX

RTPS INERTIAL SPACECRAFT STATE AT SPHERE OF
INFLUENCE

SQP TARGET CONDITION STANDARD DEVIATIONS

TCA TRAJECTORY TIME AT CLOSEST APPROACH

TSI TRAJECTORY TIME AT SPHERE OF INFLUENCE

VMCA MAGNITUDE OF VELOCITY OF VEHICLE RELATIVE
TO TARGET PLANET AT CLOSEST APPROACH

VMSI MAGNITUDE OF VELOCITY OF VEHICLE RELATIVE
TO TARGET PLANET AT SPHERE OF INFLUENCE

XCA SPACECRAFT VELOCITY RELATIVE TO TARGET
PLANET AT SPHERE OF INFLUENCE

COMMON COMPUTED/USED:	EM NO	ICL2 XP	IPRINT	ISPH	ISP2
COMMON COMPUTED:	DELTM	TRTM1			
COMMON USED:	ALNGTH	BDR	BDT	B	DATEJ
	DC	DSI	FNTM	FOV	F
	IBARY	IHYP1	ISOI1	ISTMC	NBOD
	NB	NQE	NQE	NTMC	NTP
	ONE	PHI	P	RC	RSI
	TM	VSI	ZERO		

GUIS Analysis

Subroutine GUIS is called at a midcourse guidance event at t_j in the simulation mode to compute three primary quantities for the selected midcourse guidance policy. These three quantities are the variation matrix η_j , the target condition covariance matrix prior to the velocity correction W_j , and the guidance matrix Γ_j . Three midcourse guidance policies are available: fixed-time-of-arrival (FTA), two-variable B-plane (2VBP), and three-variable B-plane (3VBP). All are linear impulsive guidance policies having form

$$\Delta \hat{V}_j = \Gamma_j \delta \hat{X}_j$$

where $\Delta \hat{V}_j$ is the commanded velocity correction, and $\delta \hat{X}_j$ is the estimate of the spacecraft position/velocity deviation from the targeted nominal. The relevant equations for each guidance policy will be summarized below.

The variation matrix η_j for FTA guidance relates deviations in spacecraft state at t_j to position deviations at time of closest approach t_{CA} , and is given by

$$\eta_j = \begin{bmatrix} \phi_1 & | & \phi_2 \end{bmatrix}$$

where $\begin{bmatrix} \phi_1 & | & \phi_2 \end{bmatrix}$ is the upper half of the state transition matrix $\Phi(t_{CA}, t_j)$. The guidance matrix for FTA guidance is given by

$$\Gamma_j = \begin{bmatrix} -\phi_2^{-1} \phi_1 & | & -I \end{bmatrix}$$

The variation matrix for 3VBP guidance relates deviations in spacecraft state at t_j to deviations in B-T, B-R, and t_{SI} , where t_{SI} is the time at which the sphere of influence is pierced. Unlike the variation matrix for FTA guidance, which can be computed analytically or by numerical differencing, the 3VBP variation matrix must always be computed using numerical differencing since no good analytical formulas are available which relate deviations in spacecraft state at t_j to deviations in t_{SI} . If the variation matrix is written as

$$\eta_j = \begin{bmatrix} \eta_1 & | & \eta_2 \end{bmatrix}$$

then the guidance matrix for 3VBP guidance is given by

$$\Gamma_j = \begin{bmatrix} -\eta_2^{-1} \eta_1 & | & -I \end{bmatrix}$$

The variation matrix for 2VBP guidance relates deviations in spacecraft state at t_j to deviations in B·T and B·R and is given by

$$\eta_j = M \Phi(t_{SI}, t_j)$$

where M is an analytically computed matrix relating B·T and B·R deviations to spacecraft state deviations at t_{SI} , and $\Phi(t_{SI}, t_j)$ is the state transition matrix over $[t_j, t_{SI}]$. If η_j is written as

$$\eta_j = \begin{bmatrix} A & | & B \end{bmatrix}$$

then the guidance matrix for 2VBP guidance is given by

$$\Gamma_j = \begin{bmatrix} -B^T (BB^T)^{-1} A & | & -B^T (BB^T)^{-1} B \end{bmatrix}$$

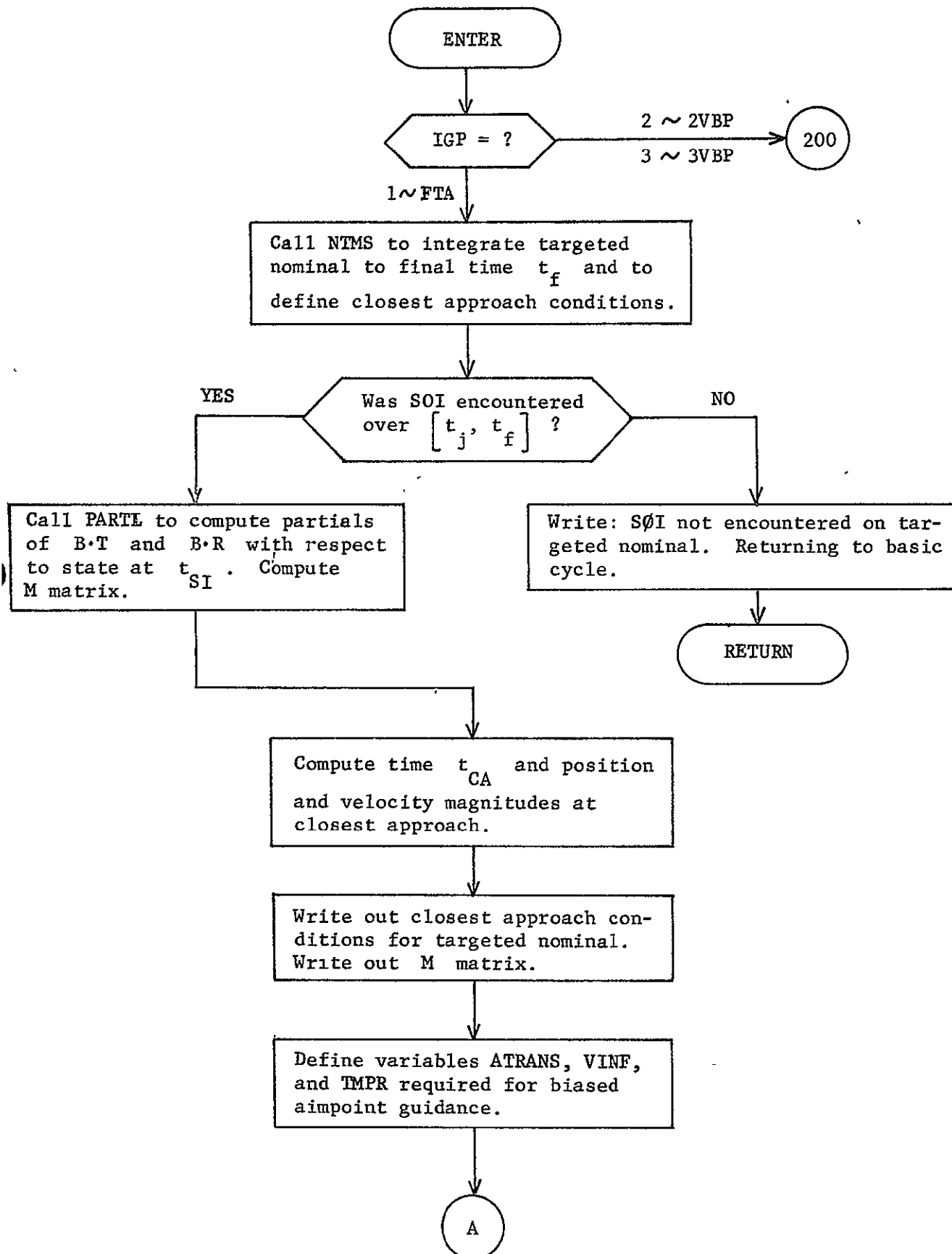
All state transition matrices and, hence, all variation matrices used by the above three guidance policies are referenced to the most recent nominal trajectory for improved numerical accuracy.

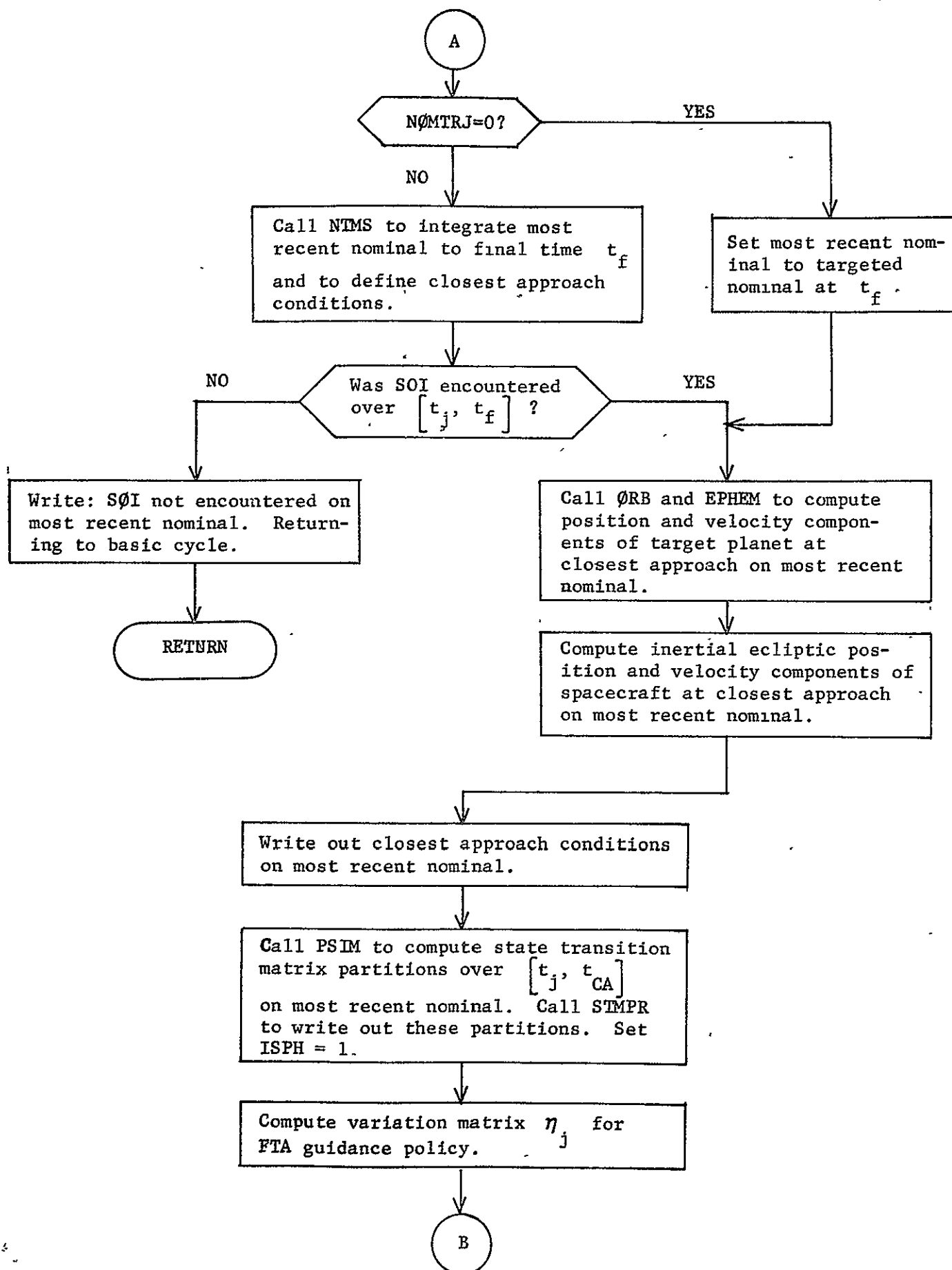
Once the variation matrix η_j is available for any of the above guidance policies, the target condition covariance matrix can be computed using

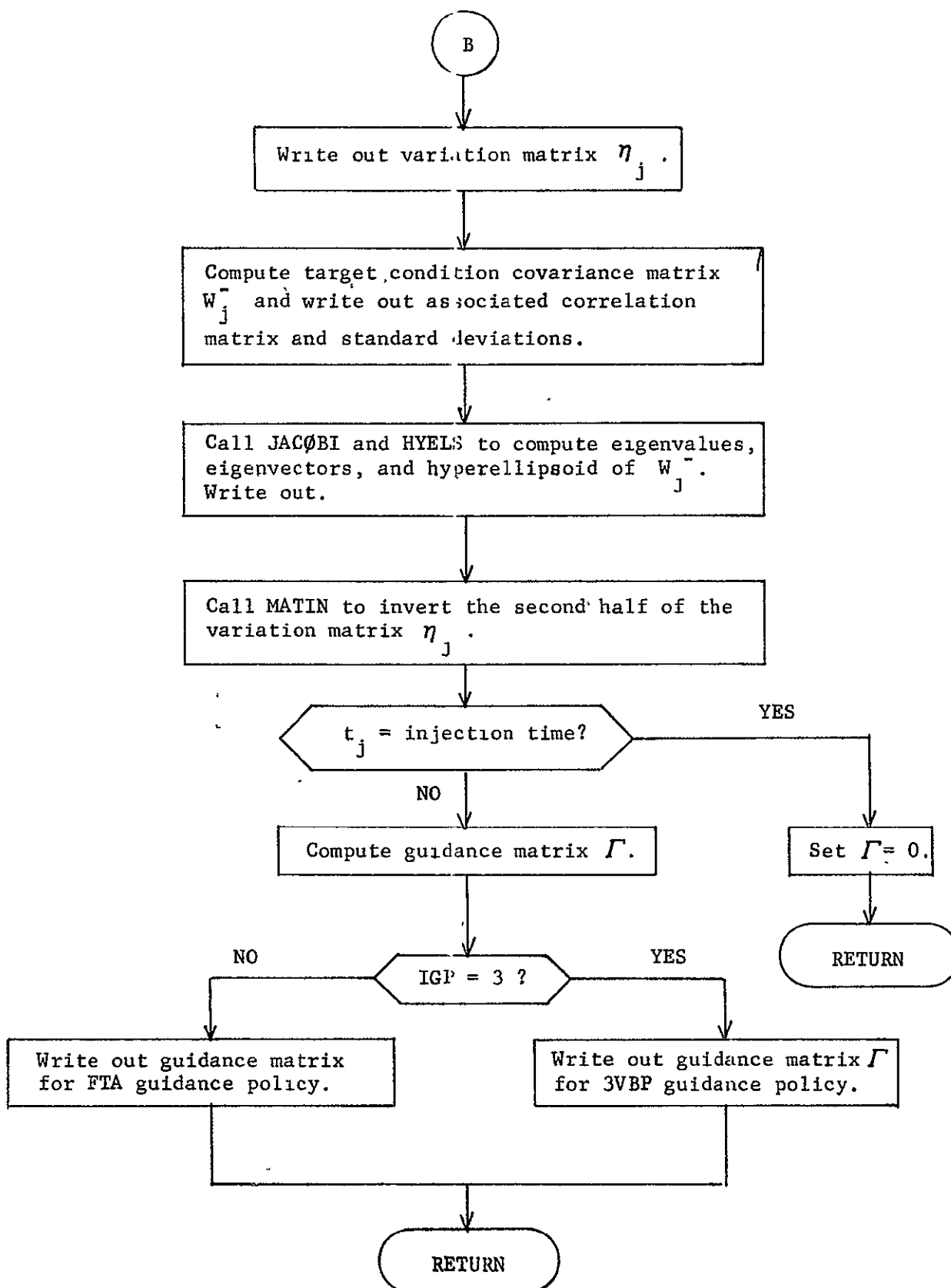
$$W_j^- = \eta_j P_{c_j}^- \eta_j^T$$

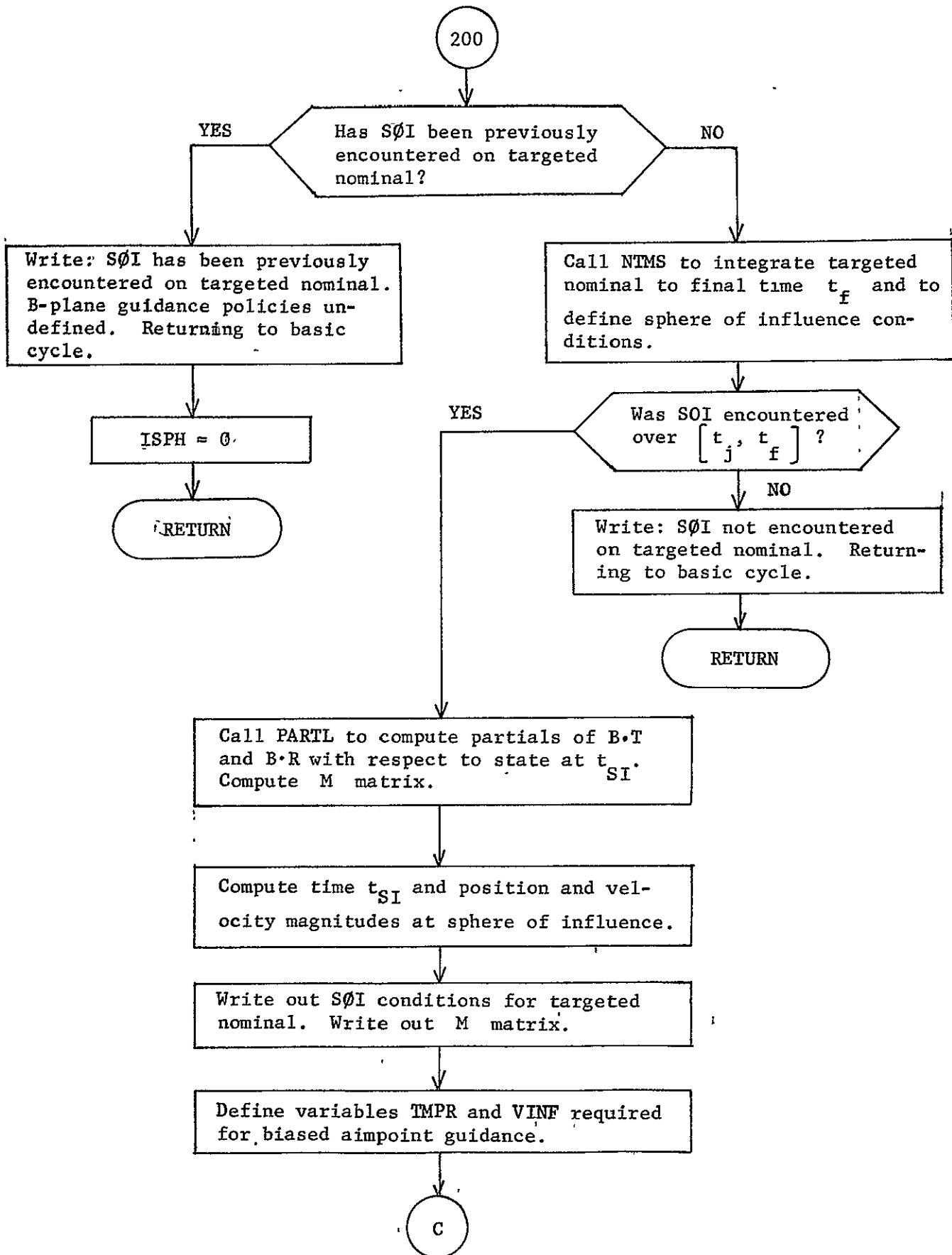
where $P_{c_j}^-$ is the control covariance matrix immediately prior to the guidance event.

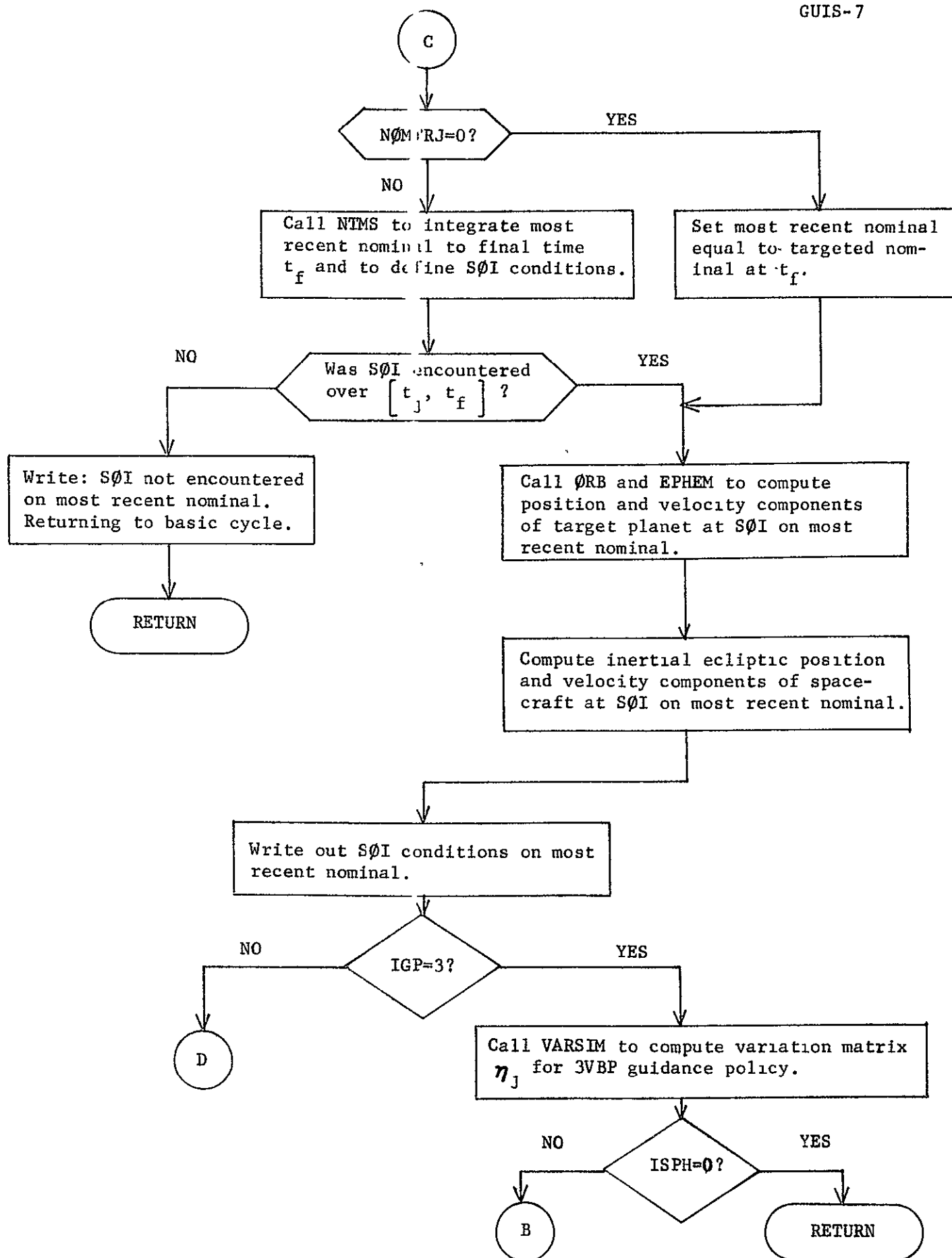
GUIS Flow Chart

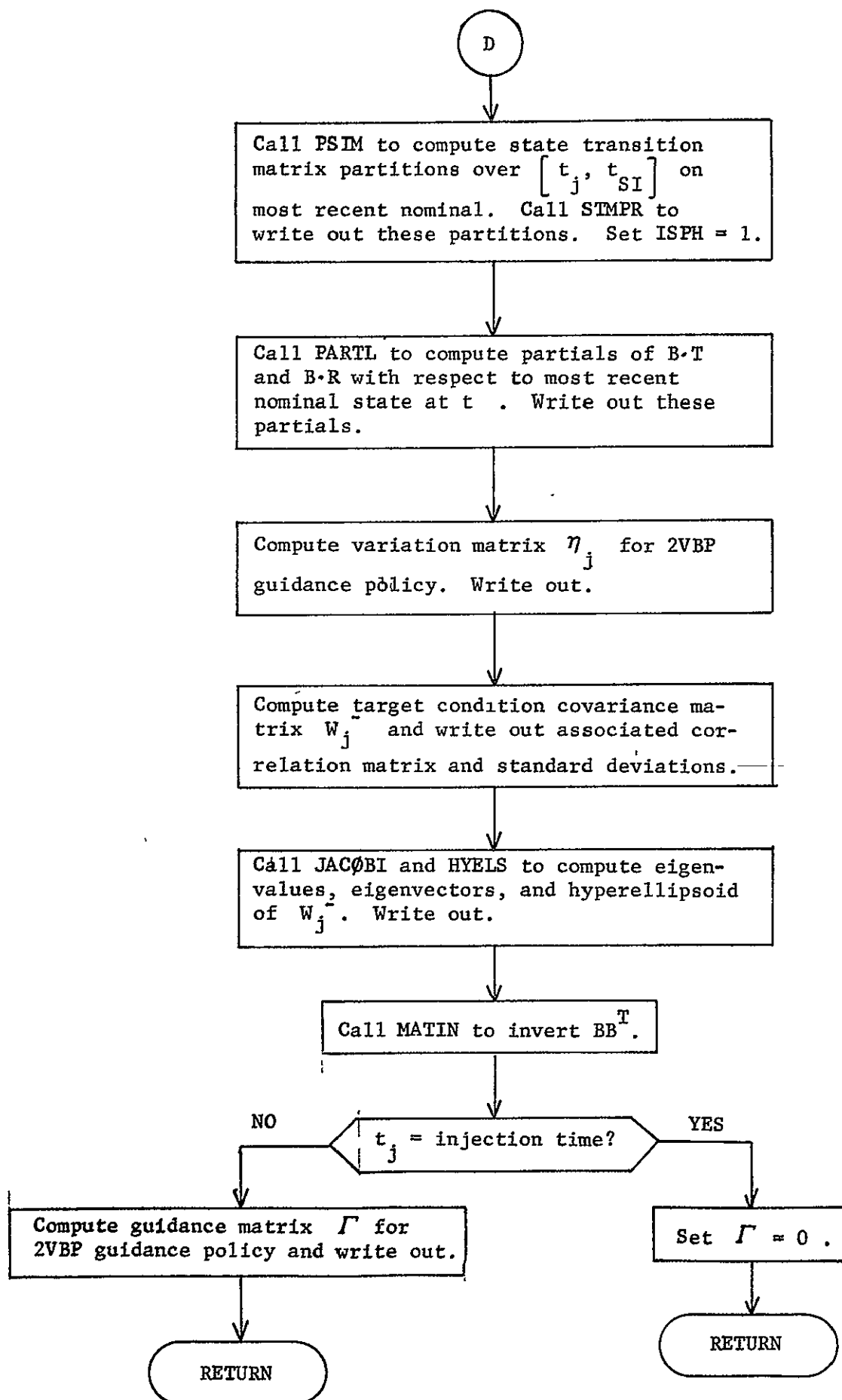












PROGRAM GUISIM

PURPOSE CONTROL EXECUTION OF GUIDANCE EVENT IN THE SIMULATION
PROGRAM

SUBROUTINES SUPPORTED: SIMULL

SUBROUTINES REQUIRED:	CORREL	DYNOS	GUIS	HYELS	JACOBI
	NAVM	PSIM	STMPR		

LOCAL SYMBOLS:	ADA	VARIATION MATRIX
	AK1	ACTUAL RESOLUTION ERROR
	AL1	ACTUAL ERROR IN POINTING ANGLE ALPHA
	BT1	ACTUAL ERROR IN POINTING ANGLE BETA
	CXSU1	STORAGE FOR CXSU KNOWLEDGE COVARIANCE
	CXSV1	STORAGE FOR CXSV KNOWLEDGE COVARIANCE
	CXU1	STORAGE FOR CXU KNOWLEDGE COVARIANCE
	CXV1	STORAGE FOR CXV KNOWLEDGE COVARIANCE
	CXXS1	STORAGE FOR CXXS KNOWLEDGE COVARIANCE
	DELX	ESTIMATED STATE DEVIATION FROM TARGETED NOMINAL TRAJECTORY
	DUM1	INTERMEDIATE VARIABLE
	DUM2	ARRAY OF EIGENVECTORS
	DVCM	MAGNITUDE OF COMMANDED MIDCOURSE VELOCITY CORRECTION
	DVC	COMMANDED MIDCOURSE VELOCITY CORRECTION
	DVE	ERROR IN MIDCOURSE VELOCITY CORRECTION DUE TO NAVIGATION UNCERTAINTY
	DV	PERFECT MIDCOURSE VELOCITY CORRECTION
	DX	ACTUAL STATE DEVIATION FROM TARGETED NOMINAL TRAJECTORY
	EGVCT	ARRAY OF EIGENVECTORS
	EGVL	ARRAY OF EIGENVALUES
	EXEC	EXECUTION ERROR COVARIANCE MATRIX

EXM MAGNITUDE OF UPDATE VELOCITY CORRECTION

GAP INTERMEDIATE ARRAY EQUAL TO GA TIMES P

GA GUIDANCE MATRIX

ICODE2 INTERNAL CONTROL FLAG

IGP MIDCOURSE GUIDANCE POLICY CODE

OUT SPACECRAFT VELOCITY RELATIVE TO TARGET
PLANET IN PLANETO-CENTRIC EQUATORIAL
COORDINATES

PS1 STORAGE FOR PS KNOWLEDGE COVARIANCE

P1 STORAGE FOR P KNOWLEDGE COVARIANCE

RF1 MOST RECENT NOMINAL SPACECRAFT STATE AT
GUIDANCE EVENT

RF TARGETED NOMINAL SPACECRAFT STATE AT
GUIDANCE EVENT

ROW INTERMEDIATE VECTOR

SQP INTERMEDIATE VECTOR

S1 ACTUAL PROPORTIONALITY ERROR

VEIG MATRIX TO BE DIAGONALIZED

Z INTERMEDIATE ARRAY

COMMON COMPUTED/USED:

ADEVX	CXSUG	CXSU	CXSVG	CXSV
CXUG	CXU	CXVG	CXV	CXXSG
CXXS	EDEVX	ICODE	NGE	PG
PSG	PS	P	RI1	TG
XF1	XG			

COMMON COMPUTED:

DELTN	TRTN1	XI1	XI
-------	-------	-----	----

COMMON USED:

AALP	ABET	ADEVXS	APRO	ARES
EDEVXS	FOP	FOV	ICDT3	IEIG
IHYP1	ISPH	ISTMC	NDIM1	NDIM2
NDIM3	Q	SIGALP	SIGBET	SIGPRO
SIGRES	TEVN	U0	V0	W
XF	XSL	ZERO		

GUISIM Analysis

Subroutine GUISIM is the executive guidance subroutine in the simulation program. In addition to controlling the computational flow for all types of guidance events, GUISIM also performs many of the required guidance computations itself.

Before considering each type of guidance event, the treatment of a general guidance event will be discussed. Let t_j be the time at which the guidance event occurs. Before any guidance event can be executed the targeted nominal state \bar{X}_j , most recent nominal state \tilde{X}_j , estimated state deviation $\delta\tilde{X}_j$ from most recent nominal, actual state deviation $\delta\tilde{X}_j$ from most recent nominal, knowledge covariance $P_{K_j}^-$, and control covariance $P_{C_j}^-$ must all be available, where $()^-$ indicates values immediately before the event. Only the control covariance is not available prior to entering GUISIM. The propagation of the control covariance over the interval $[t_{j-1}, t_j]$, where t_{j-1} denotes the time of the previous guidance event, is performed within GUISIM.

The next step in the treatment of a general guidance event is concerned with the computation of the commanded velocity correction, execution error covariance, actual execution error, and actual velocity correction. In the simulation program a non-statistical commanded velocity correction can always be computed. This commanded velocity correction $\Delta\hat{V}_j$ is used to compute the execution error covariance matrix \tilde{Q}_j and the actual execution error $\delta\Delta V_j$. A summary of the execution error model and the equations used to compute \tilde{Q}_j and $\delta\Delta V_j$ can be found in the subroutine QCØMP analysis section. The actual velocity correction is then computed using the equation

$$\Delta V_j = \Delta\hat{V}_j + \delta\Delta V_j$$

The last step is concerned with the updating of required quantities prior to returning to the basic cycle. An assumption underlying the modeled guidance process is that the targeted nominal is always updated by the commanded velocity correction. In the simulation program the update velocity correction $\Delta\hat{V}_{UP_j}$ is always identical to the commanded velocity correction $\Delta\hat{V}_j$. This is in contrast to the error analysis program where $\Delta\hat{V}_{UP_j}$ is equated with the non-statistical component of $\Delta\hat{V}_j$. The

most recent and targeted nominal states immediately following the guidance event are updated using the equations

$$\tilde{\mathbf{x}}_j^+ = \tilde{\mathbf{x}}_j^- + \delta \tilde{\mathbf{x}}_j^- + \begin{bmatrix} 0 \\ -\frac{\delta \tilde{\Delta V}_{UP_j}}{\delta \tilde{\Delta V}_{UP_j}} \end{bmatrix}$$

$$\bar{\mathbf{x}}_j^+ = \tilde{\mathbf{x}}_j^+$$

The actual and estimated state deviations from the most recent nominal are given by

$$\delta \tilde{\mathbf{x}}_j^+ = \delta \tilde{\mathbf{x}}_j^- - \delta \tilde{\mathbf{x}}_j^- + \begin{bmatrix} 0 \\ -\delta \tilde{\Delta V}_j^- \end{bmatrix}$$

$$\delta \tilde{\mathbf{x}}_j^+ = 0$$

The previous 4 equations assume an impulsive thrust model. If, instead, the thrust is modeled as an impulse series, then an effective estimated state $\hat{\mathbf{x}}_{eff}$ and an effective actual state $\hat{\mathbf{x}}_{eff}$ are computed.

The equations used to compute these effective states are summarized in the subroutine PULSEX analysis section. The previous update equations are then replaced by the following equations

$$\tilde{\mathbf{x}}_j^+ = \hat{\mathbf{x}}_{eff}$$

$$\bar{\mathbf{x}}_j^+ = \tilde{\mathbf{x}}_j^+$$

$$\delta \tilde{\mathbf{x}}_j^+ = \mathbf{x}_{eff} - \hat{\mathbf{x}}_{eff}$$

$$\delta \hat{\mathbf{x}}_j^+ = 0$$

The knowledge covariance is updated using the equation

$$\mathbf{P}_{K_j}^+ = \mathbf{P}_{K_j}^- + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \tilde{\mathbf{Q}}_j \end{bmatrix}$$

if an impulsive thrust model is assumed. If the thrust is modeled as a series of impulses, then an effective execution error covariance \tilde{Q}_{eff} is computed and the knowledge covariance is updated using the equation

$$P_{Kj}^+ = P_{Kj}^- + \tilde{Q}_{eff}$$

In either case the control covariance is updated simply by setting

$$P_{cj}^+ = P_{Kj}^+$$

This equation is a direct consequence of the assumption that the targeted nominal is always updated at a guidance event.

A "compute only" option is available in GUISIM in which all of the $()^+$ quantities will still be computed and printed. However, states, deviations, and covariances are then reset to their former $()^-$ values prior to returning to the basic cycle.

Each specific type of guidance event involves the computation of other quantities not discussed above. These will be covered in the following discussion of specific guidance events.

1. Midcourse and biased aimpoint guidance.

Linear midcourse guidance policies have form

$$\Delta \hat{V}_{Nj} = \Gamma_j \delta \hat{x}_j$$

where the subscript N indicates that this is the velocity correction required to null out deviations from the nominal target state. This notation is required to differentiate between this type of velocity correction and velocity corrections required to achieve an altered target state. Linear midcourse guidance policies are discussed in more detail in the subroutine GUI analysis section.

Subroutine GUISIM calls GUI to compute the guidance matrix, Γ_j , and the target condition covariance immediately prior to the guidance event, W_j^- , and then uses Γ_j to compute the velocity correction covariance S_j , which is defined as

$$S_j = E [\hat{\Delta V}_{N_j} \hat{\Delta V}_{N_j}^T],$$

and is given by the equation

$$S_j = \Gamma_j (P_{c_j}^- - P_{K_j}^-) \Gamma_j^T$$

This equation assumes that an optimal estimation algorithm is employed in the navigation process, since the derivation of this equation requires the orthogonality of the estimate and the estimation error.

Since state estimates $\hat{\delta X}_j$ are generated in the simulation program, an actual $\hat{\Delta V}_{N_j}$ can always be computed. This is in contrast to the error analysis program where only a statistical or effective $\hat{\Delta V}_{N_j}$ can be computed.

The perfect velocity correction $\underline{\Delta V}_j$, defined as the velocity correction required to null out actual deviations from the nominal target state, is also computed for midcourse guidance events. Assuming linear guidance theory, the perfect velocity correction is given by

$$\underline{\Delta V}_j = \Gamma_j \delta X_j$$

where δX_j is the actual deviation from the targeted nominal. An option is also available in GUISIM for re-computing $\hat{\Delta V}_{N_j}$ using nonlinear techniques.

However, it should be noted that the nonlinear two-variable B-plane guidance policy, unlike the corresponding linear policy, constrains the z-component of $\hat{\Delta V}_{N_j}$ to be zero.

If planetary quarantine constraints must be satisfied at a midcourse correction, GUISIM calls BIAIM to compute the new aimpoint μ_j and the bias velocity correction $\hat{\Delta V}_{B_j}$. All computations in BIAIM are based on

linear guidance theory. However, an option is available in GUISIM to re-compute the total velocity correction $\hat{\Delta V}_{B_j} + \hat{\Delta V}_{N_j}$, but not μ_j , using non-

linear techniques. This option is recommended if a biased aimpoint guidance event occurs at t_j = injection time. It should also be noted that \hat{Q}_j is set to zero if t_j = injection time since it is assumed that the injection

covariance does not change for small changes in injection velocity.

After the updated control covariance P_c^+ has been computed, the target condition covariance matrix W_j^+ following the guidance correction is computed using the equation

$$W_j^+ = \eta_j P_c^+ \eta_j^T$$

where variation matrix η_j has been previously computed in subroutine GUIS.

2. Re-targeting.

In the simulation (and error analysis) program a re-targeting event is defined to be the computation of a velocity correction $\Delta \hat{V}_{RT}$ required to achieve a new set of target conditions using nonlinear techniques. Since the state estimate $\tilde{X}_j^- + \delta \tilde{X}_j^-$ is used as the zero-th iterate in the re-targeting process, the new target conditions must be close enough to the original nominal target conditions to ensure a convergent process.

It should be noted that after a re-targeting event the new target conditions are henceforth treated as the nominal target conditions.

3. Orbital insertion.

An orbital insertion event is divided into a decision event and an execution event. At a decision event the orbital insertion velocity correction $\Delta \hat{V}_{OI}$ and the time interval Δt separating decision and execution are computed based on the state estimate $\tilde{X}_j^- + \delta \tilde{X}_j^-$. The relevant equations can be found in the subroutine COPINS analysis section for coplanar orbital insertion; in NOPINS, for non-planar orbital insertion. Before returning to the basic cycle, GUISISIM schedules the orbital insertion execution event to occur at $t_j + \Delta t$ and re-orders the necessary event arrays accordingly.

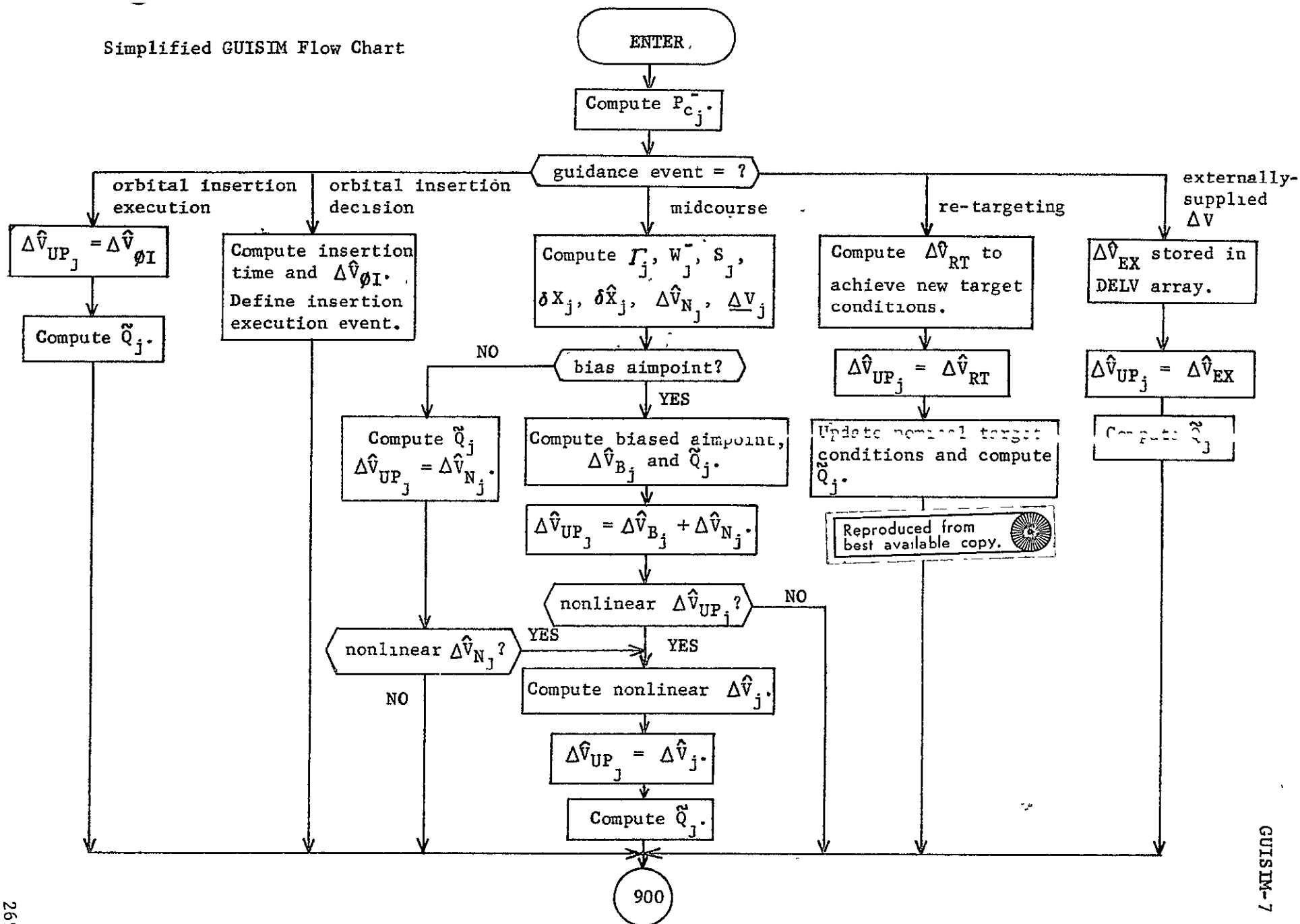
At an orbital insertion execution event the previously computed $\Delta \hat{V}_{OI}$ is used to update the targeted nominal state. In addition, the planeto-centric equatorial components of $\Delta \hat{V}_{OI}$ and the actual spacecraft cartesian and orbital element states following the insertion maneuver are computed.

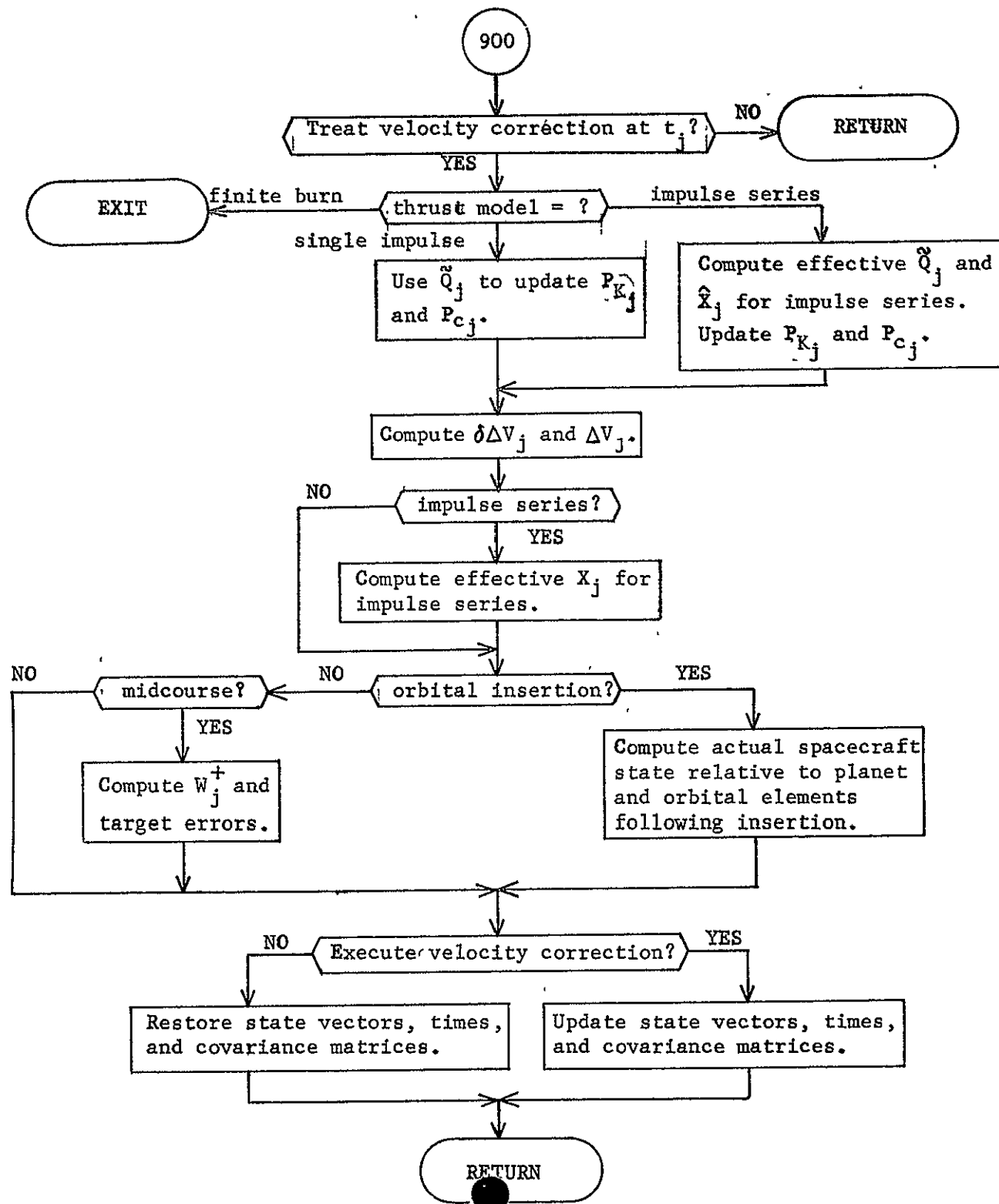
4. Externally-supplied velocity correction.

At this type of guidance event the state estimate $\tilde{\mathbf{X}}_j + \hat{\tilde{\mathbf{X}}}_j$ is simply updated using the externally-supplied velocity correction $\Delta \hat{\mathbf{V}}_{EX}$.

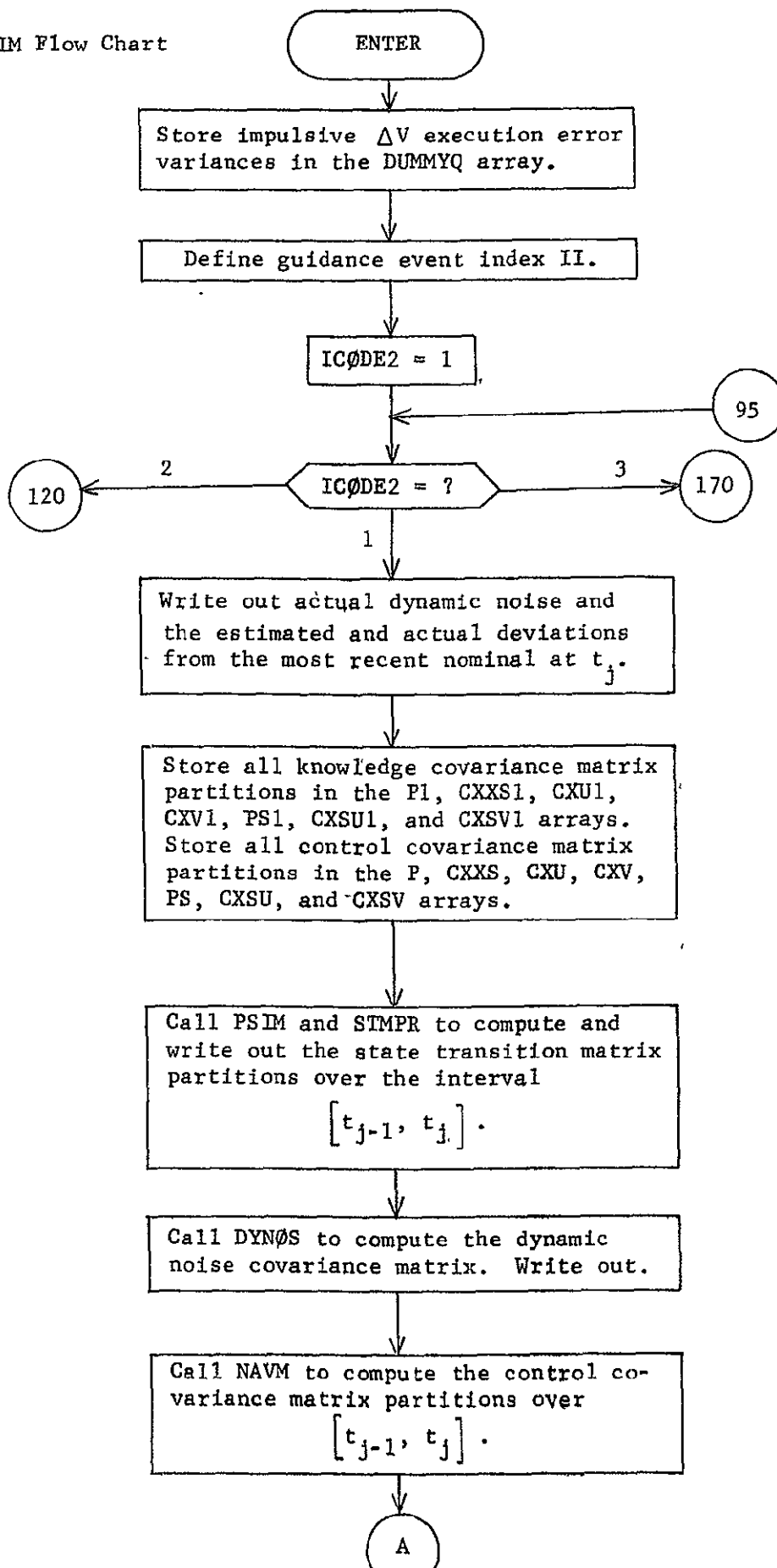
Because of the complexity of the GUISIM flow chart, a simplified flow chart depicting the main elements of the GUISIM structure precedes the complete GUISIM flow chart.

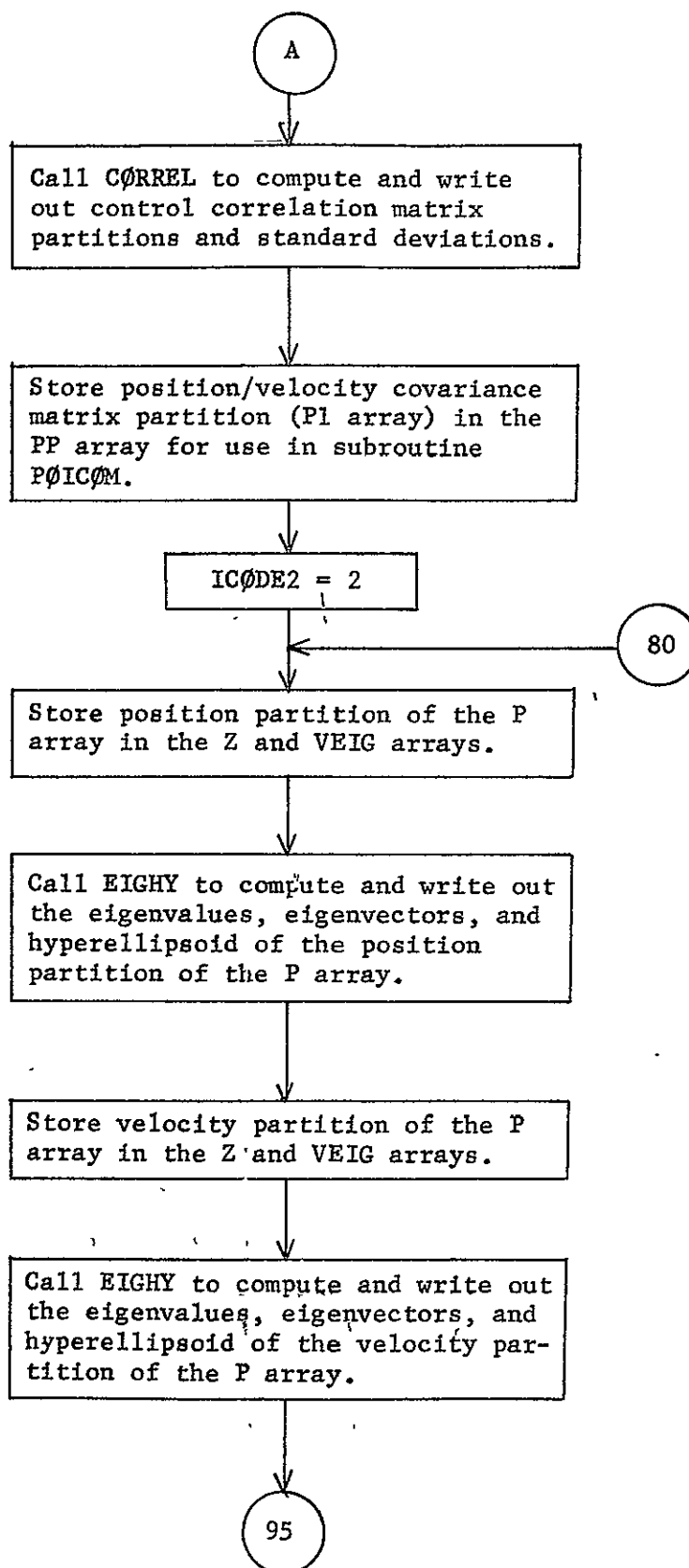
Simplified GUISIM Flow Chart

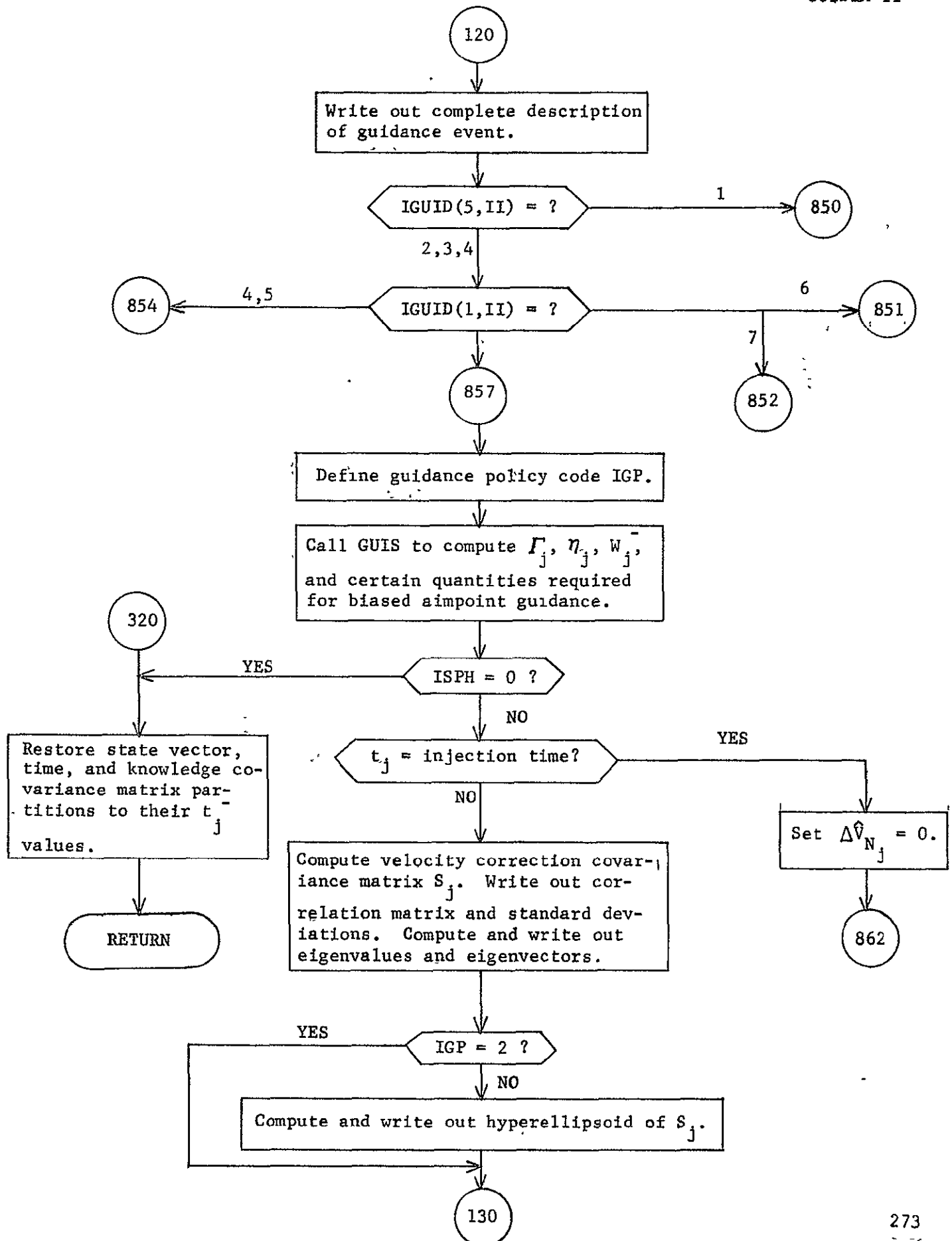


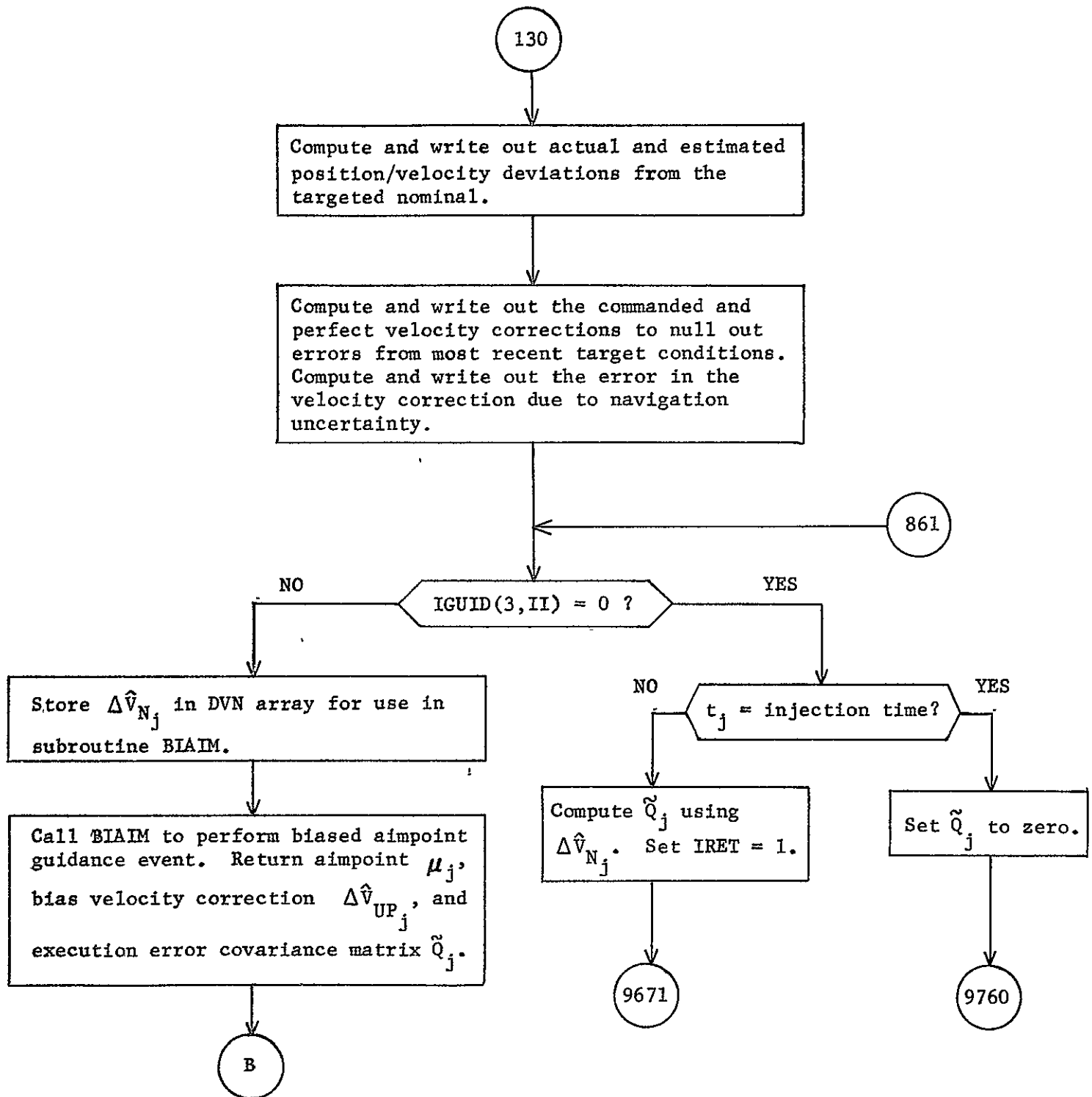


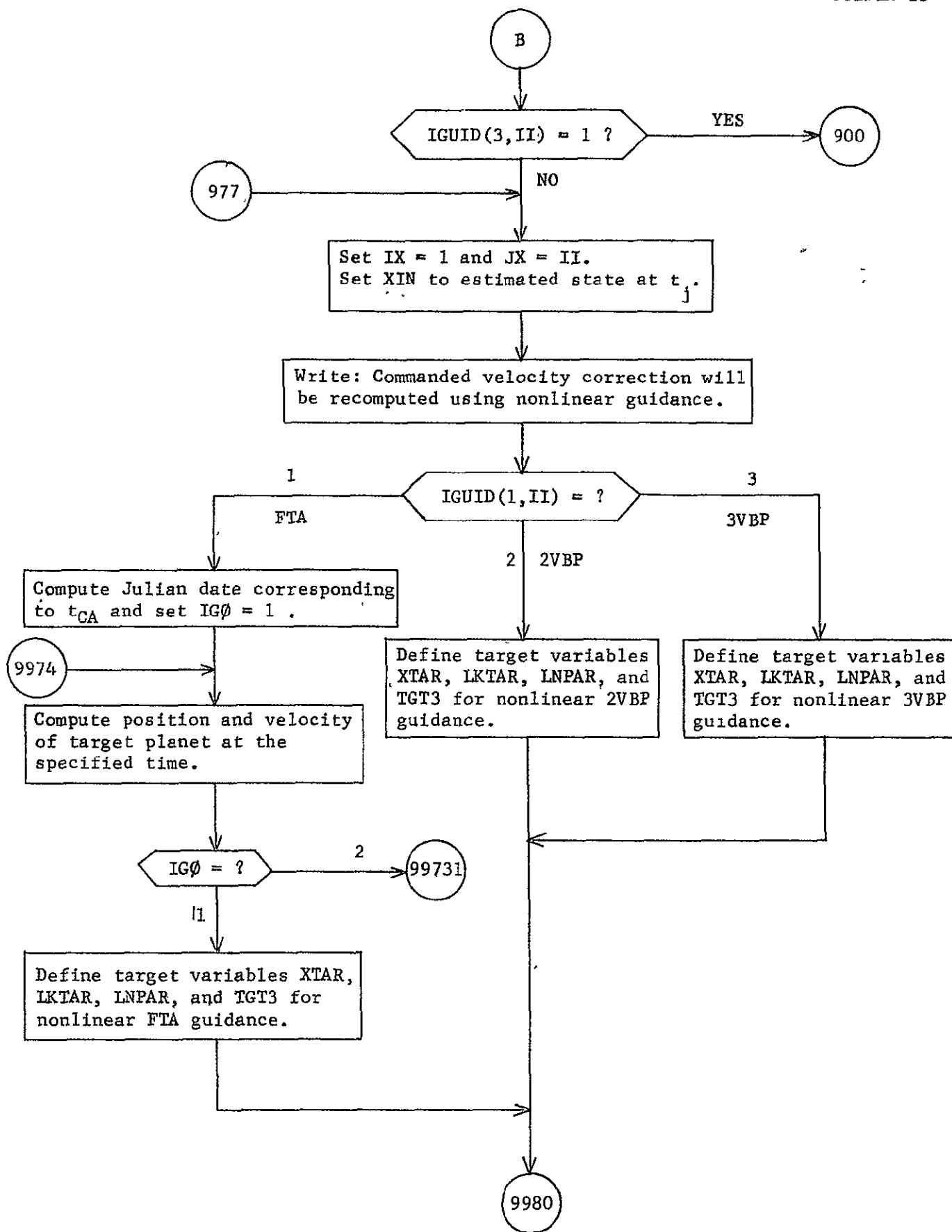
GUISIM Flow Chart

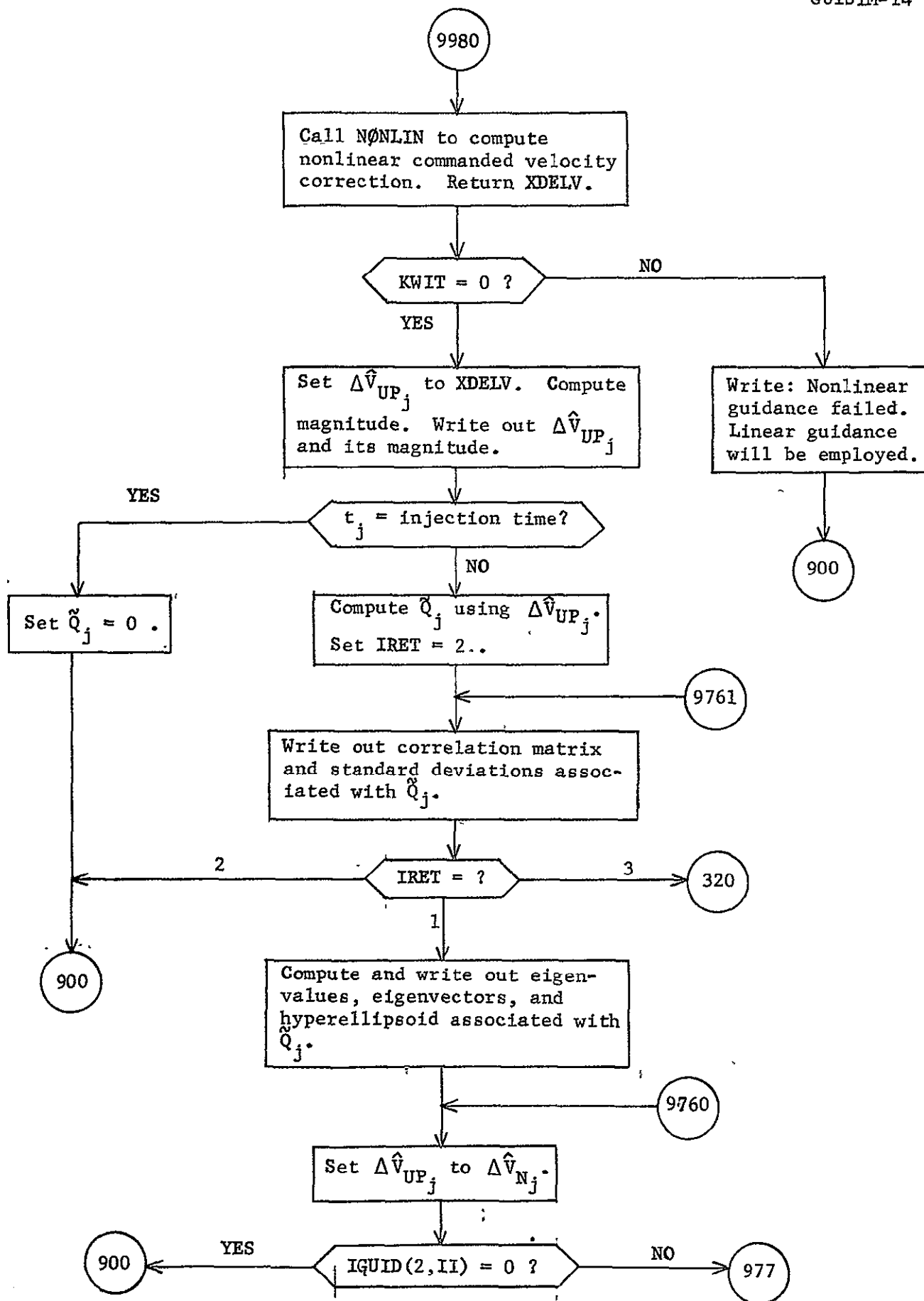


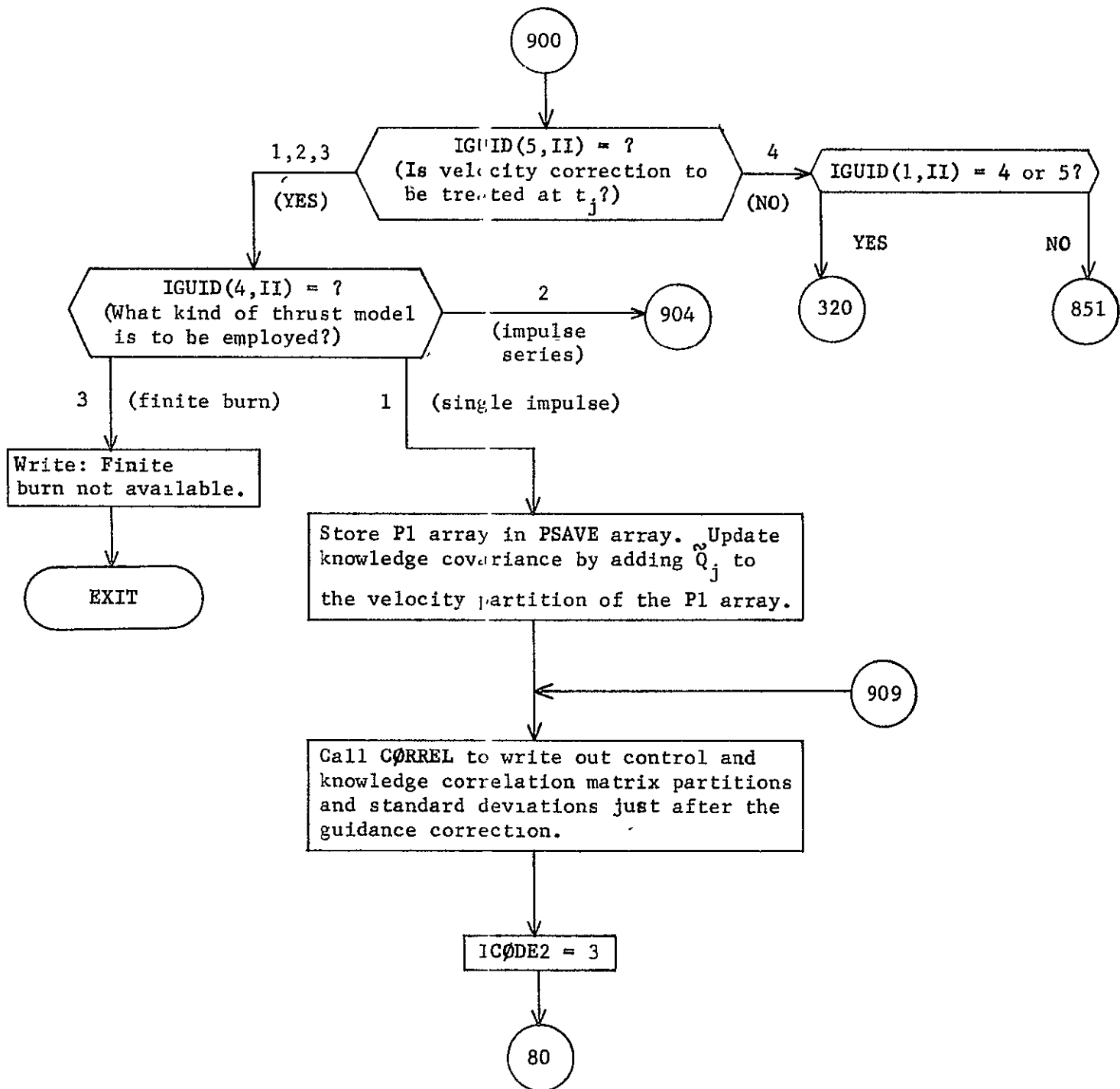


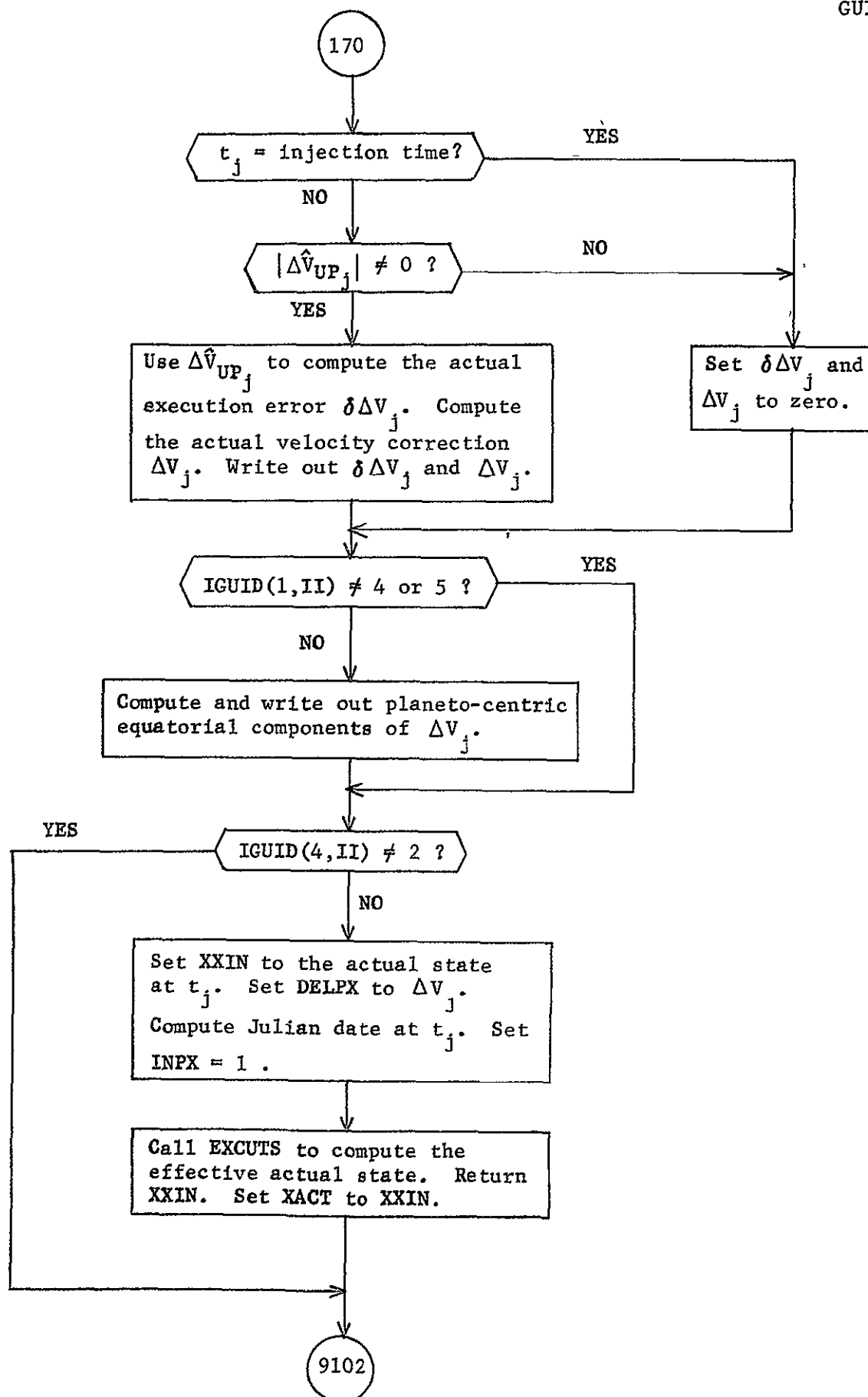


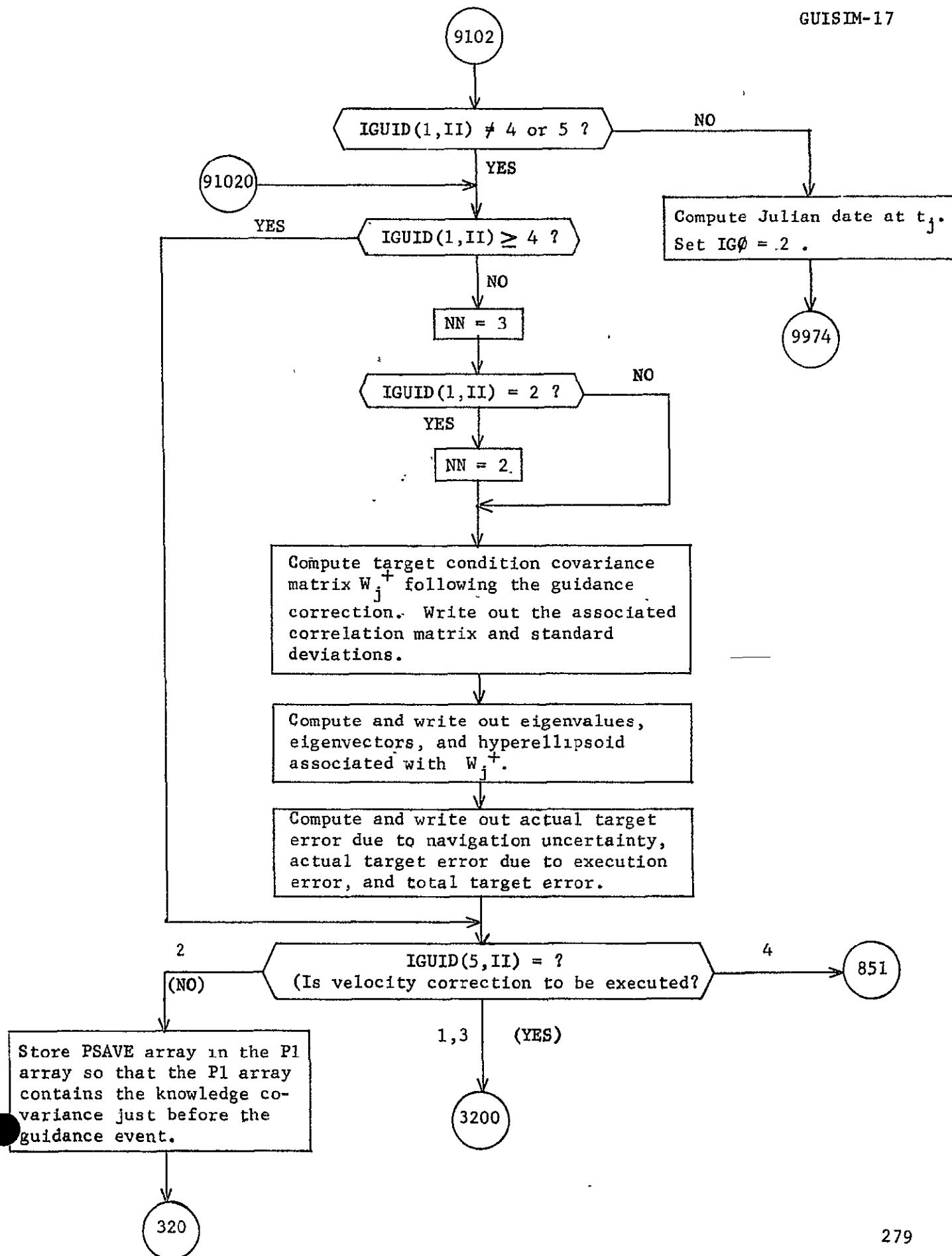


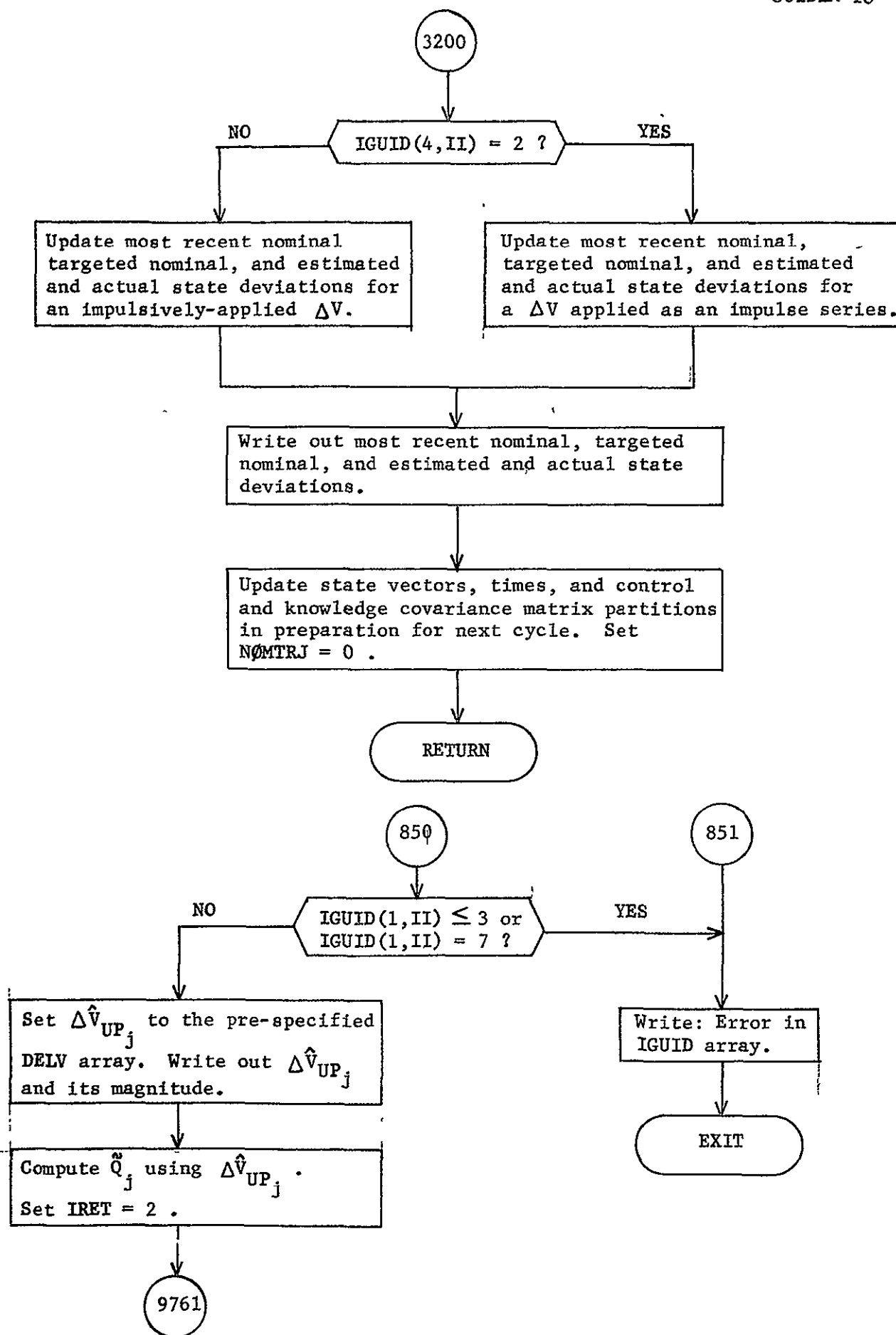


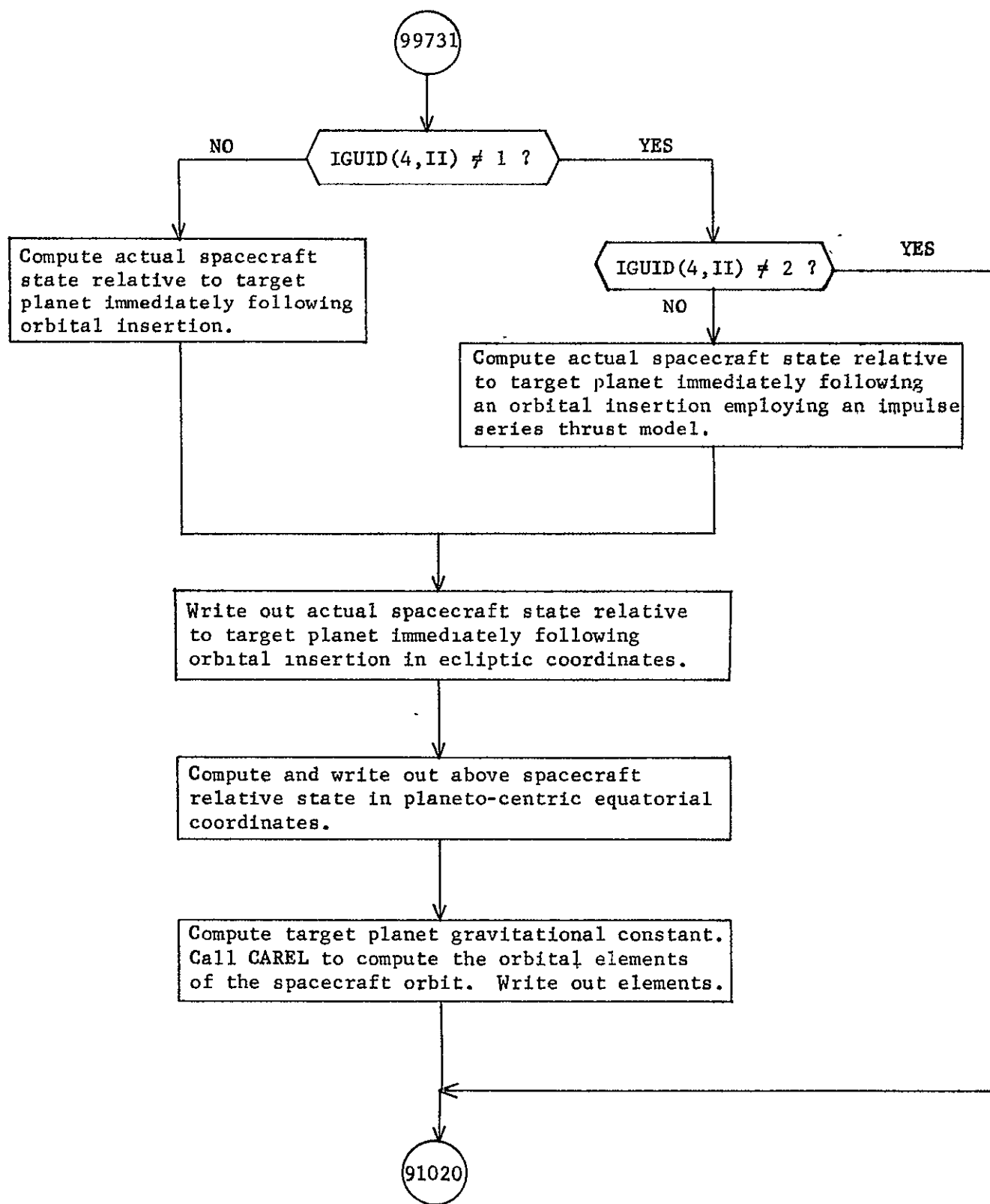


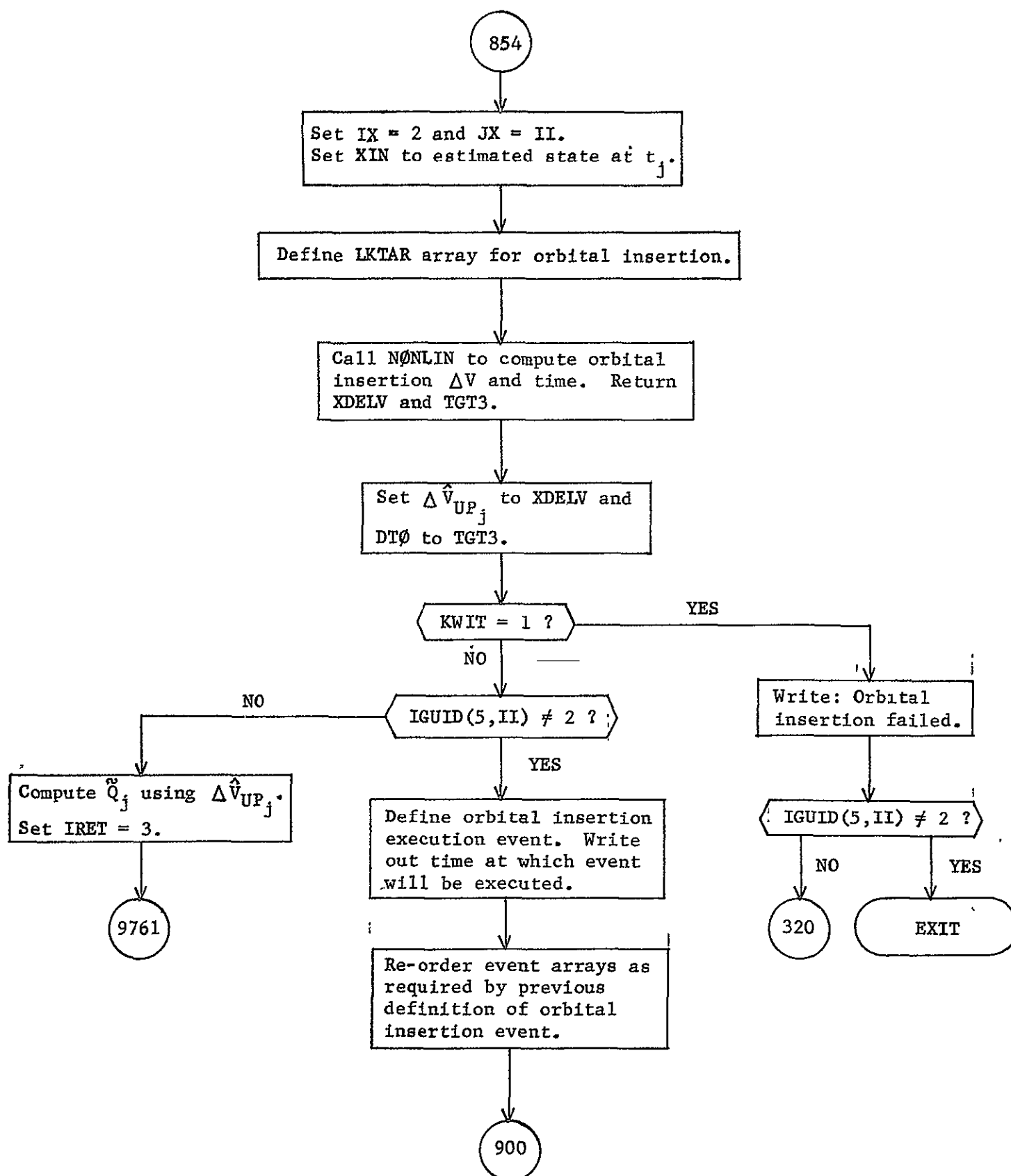


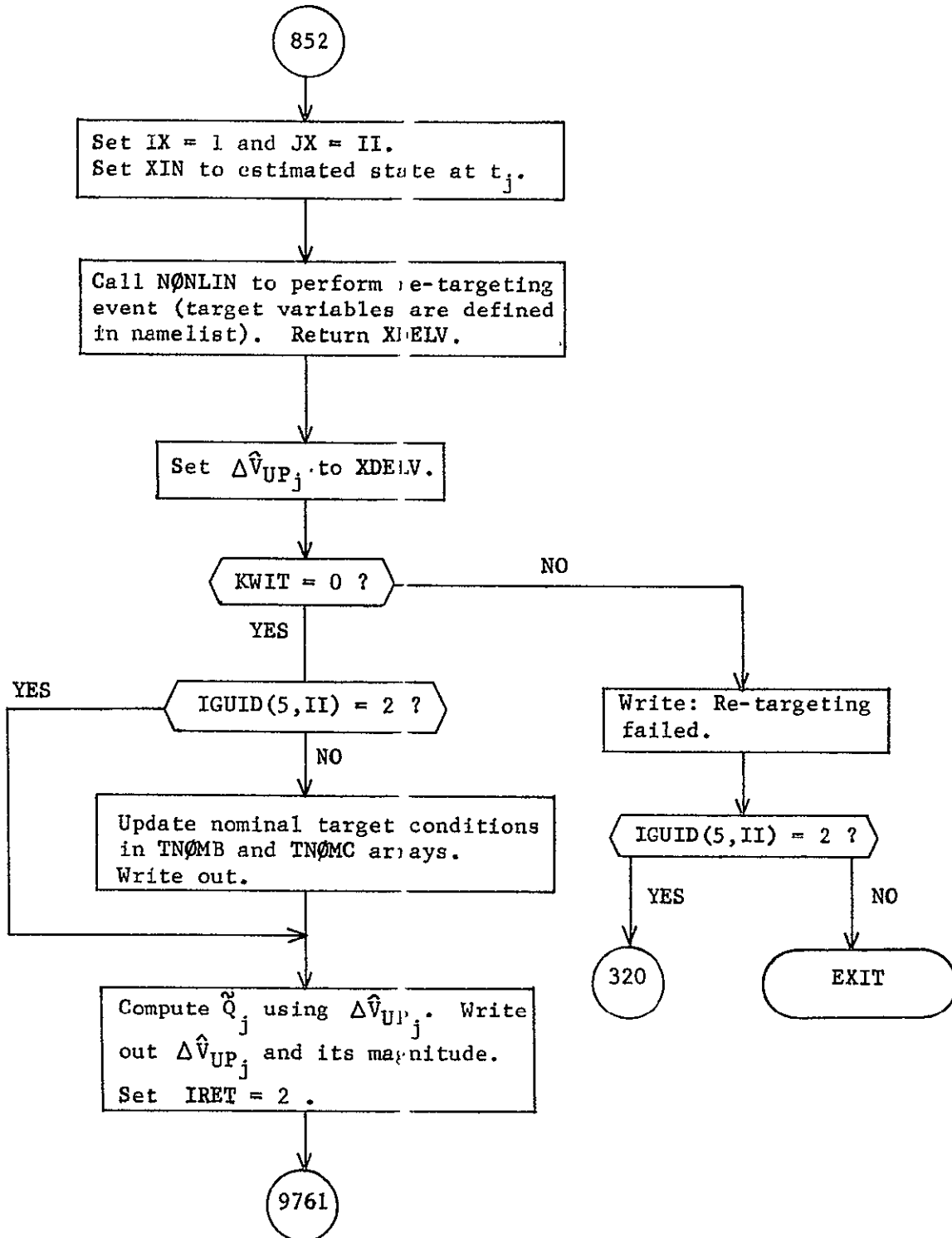


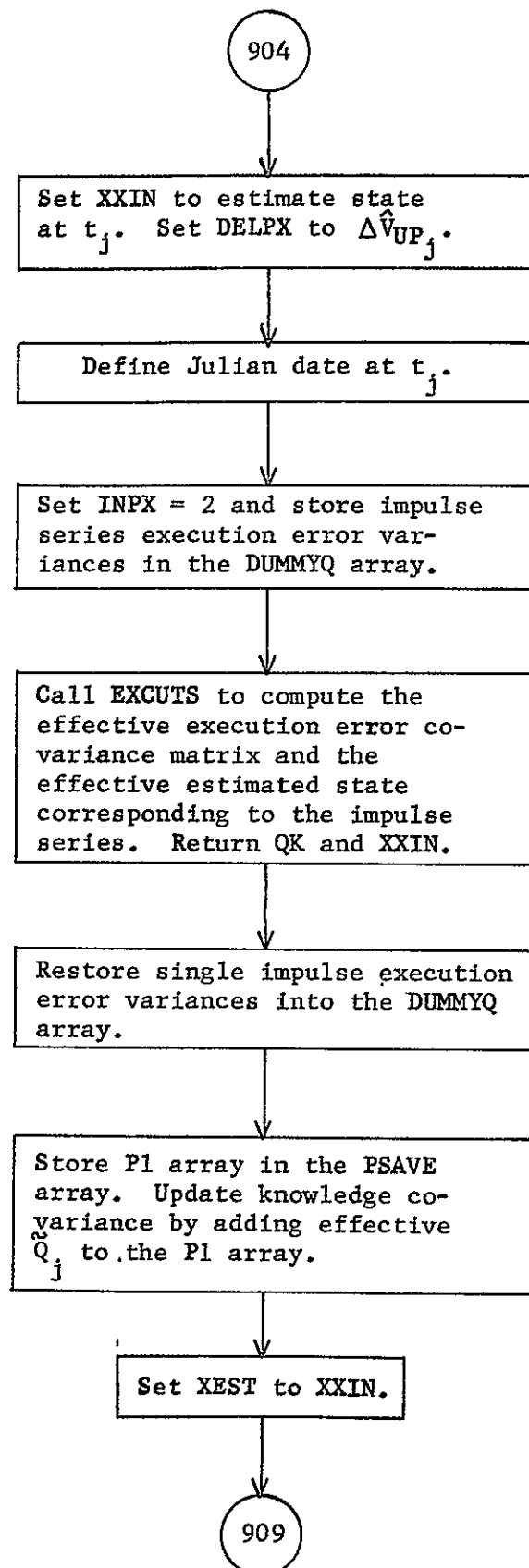












SUBROUTINE HELIO

PURPOSE: TO COMPUTE THE ZERO ITERATE INJECTION STATE FOR
INTERPLANETARY TARGETING

CALLING SEQUENCE: CALL HELIO

SUBROUTINES SUPPORTED: ZERIT

SUBROUTINES REQUIRED: LAUNCH FLITE ELCAR EPHEM ORB
PECEQ TIME

LOCAL SYMBOLS: AHEL SEMI-MAJOR AXIS OF THE HELIOCENTRIC CONIC

ARGP ARGUMENT OF PERIAPSIS OF THE HELIOCENTRIC
CONIC IN RADIAN

ASCND LONGITUDE OF THE ASCENDING NODE OF THE
HELIOCENTRIC CONIC IN RADIAN

ATP SEMI-MAJOR AXIS OF TARGET PLANETOCENTRIC
CONIC

AZF AZIMUTH AT DF ON THE HELIOCENTRIC CONIC
IN DEGREES

AZI AZIMUTH AT DI ON THE HELIOCENTRIC CONIC
IN DEGREES

B2 SQUARE OF THE B VECTOR MAGNITUDE OF THE
TARGET PLANETOCENTRIC CONIC

CBDR DESIRED B.R MAGNITUDE AT DF OF THE TARGET
PLANETOCENTRIC CONIC

CBDT DESIRED B.T MAGNITUDE AT DF OF THE TARGET
PLANETOCENTRIC CENTER

COSASN COSINE OF ASCND

COSB INTERMEDIATE VARIABLE FOR AZI,AZF EQUATION

COSFS COSINE OF FS

COSF COSINE OF TAI

COSPSI COSINE OF PSI

COSTHE COSINE OF THETA

CRCA DESIRED RCA MAGNITUDE AT DF OF THE TARGET
PLANETOCENTRIC CONIC

DELT	TIME OF FLIGHT (SECS) OF HELIOCENTRIC CONIC
DF	FINAL JULIAN DATE OF HELIOCENTRIC CONIC
DI	INITIAL JULIAN DATE OF HELIOCENTRIC CONIC
DSICA	DELTA TIME (DAYS) FROM SPHERE-OF-INFLUENCE TO CLOSEST APPROACH OF THE TARGET PLANET OCENTRIC CONIC
EHCL	ECCENTRICITY OF THE HELIOCENTRIC CONIC
EQEC	TRANSFORMATION MATRIX FROM ECLIPTIC TO TARGET PLANET EQUATORIAL AT DF
ETP	ECCENTRICITY OF THE TARGET PLANETOCENTRIC CONIC
FF	INTERMEDIATE VARIABLE FOR COMPUTATION OF DSICA
FS	TRUE ANOMALY OF THE TARGET PLANETOCENTRIC CONIC
IDAT	CALENDER DATE CORRESPONDING TO DF
IDUM	DUMMY ARGUMENT FOR CALL TO SUBROUTINE FLITE
ITIM	INDICATES COMPUTATION OF HELIOCENTRIC STATES =0, COMPUTE INITIAL AND FINAL STATES =1, COMPUTE FINAL STATE ONLY
I	INDEX
J	INDEX
LDAT	CALENDER DATE CORRESPONDING TO DI
OAPD	APOAPSIS RADIUS OF THE HELIOCENTRIC CONIC
OASN	ASCND CONVERTED TO DEGREES
OECC	OUTPUT ECCENTRICITY OF THE HELIOCENTRIC CONIC
OGAF	FLIGHT PATH ANGLE AT DF OF THE HELIOCENTRIC CONIC
OGAI	FLIGHT PATH ANGLE AT DI OF THE HELIOCENTRIC CONIC

OHCA	CENTRAL ANGLE OF THE HELIOCENTRIC CONIC
OINC	INCLINATION OF THE HELIOCENTRIC CONIC
OLAF	LATITUDE AT DF OF HELIOCENTRIC CONIC
OLAI	LATITUDE AT DI OF HELIOCENTRIC CONIC
OLOF	LONGITUDE AT DF OF HELIOCENTRIC CONIC
OLOI	LONGITUDE AT DI OF HELIOCENTRIC CONIC
OPER	ARGUMENT OF PERIAPSIS OF THE HELIOCENTRIC CONIC IN DEGREES
ORCA	PERIAPSIS RADIUS OF HELIOCENTRIC CONIC
ORF	MAGNITUDE OF HELIOCENTRIC POSITION AT DF IN OUTPUT UNITS
ORI	MAGNITUDE OF HELIOCENTRIC POSITION AT DI IN OUTPUT UNITS
OSMA	SEMI-MAJOR AXIS OF THE HELIOCENTRIC CONIC IN OUTPUT UNITS
OTAF	TRUE ANOMALY AT DF OF HELIOCENTRIC CONIC IN DEGREES
OTAI	TRUE ANOMALY AT DI OF HELIOCENTRIC CONIC IN DEGREES
OVF	MAGNITUDE OF HELIOCENTRIC VELOCITY AT DF IN KILOMETERS
OVI	MAGNITUDE OF HELIOCENTRIC VELOCITY AT DI IN KILOMETERS
OVPF	VELOCITY OF TARGET PLANET AT DF
OVPI	VELOCITY OF TARGET PLANET AT DI
PHEL	SEMI-LATUS RECTUM OF HELIOCENTRIC CONIC
PLINC	INCLINATION (IN RADIANS) OF HELIOCENTRIC CONIC
PSI	CENTRAL ANGLE (IN RADIANS) OF HELIOCENTRIC CONIC
PTP	SEMI-LATUS RECTUM OF TARGET PLANETOCENTRIC CONIC AT DF. USED TO CALCULATE DSICA

RF	MAGNITUDE OF TARGET PLANET POSITION AT DF
RI	MAGNITUDE OF TARGET PLANET POSITION AT DI
RTM	MAGNITUDE OF RT VECTOR
RT	HELIOCENTRIC POSITION VECTOR OF THE FINAL CONIC CORRESPONDING TO OTAF
RZM	MAGNITUDE OF THE RZ VECTOR
RZ	HELIOCENTRIC POSITION VECTOR OF THE FINAL CONIC CORRESPONDING TO OTAI
SGN	INTERNAL SIGN VARIABLE USED TO DEFINE THE TRANSFER PLANE ORIENTATION
SINASN	SINE OF ASCND
SINF	SIN OF TAI
SINHF	HYPERBOLIC SINE OF THE AUXILIARY VARIABLE F USED TO CALCULATE DSICA
SINPSI	SINE OF PSI
SI	SECONDS IN CALENDER DATE IDAT
SL	SECONDS IN CALENDER DATE LDAT
SUNMU	GRAVITATIONAL CONSTANT OF SUN IN KM^3/SEC^2
TAF	OTAF IN RADIANS
TAI	OTAI IN RADIANS
TANF	TANGENT OF THE AUXILIARY VARIABLE F USED TO CALCULATE DSICA
TERM	INTERMEDIATE VARIABLE USED TO CALCULATE CRCA
TEST	INTERMEDIATE VARIABLE USED TO CALCULATE AZIMUTHS AND PATH ANGLES
TFP	DUMMY VARIABLE USED TO CALL ELCAR
THETAI	INTERMEDIATE ANGLE USED TO DEFINE ARGP
TSPH	SPHERE-OF-INFLUENCE OF TARGET PLANET IN KILOMETERS

VF	VELOCITY OF THE TARGET PLANET AT DF				
VHAT	INTERMEDIATE VECTOR USED TO DEFINE AZI, AZF				
VHP	HYPERBOLIC EXCESS VELOCITY OF THE TARGET PLANETOCENTRIC CONIC AT DF				
VHPM	MAGNITUDE OF THE VHP VECTOR USED TO CALCULATE DSICA				
VI	VELOCITY OF LAUNCH PLANET AT DI				
VMAG	INTERMEDIATE VARIABLE USED TO DEFINE AZI, AZF				
VTM	MAGNITUDE OF VT VECTOR				
VT	HELIOCENTRIC VELOCITY VECTOR OF THE FINAL CONIC CORRESPONDING TO OTAF				
VZM	MAGNITUDE OF THE VZ VECTOR				
VZ	HELIOCENTRIC VELOCITY VECTOR OF THE FINAL CONIC CORRESPONDING TO OTAI				
WHAT	UNIT VECTOR NORMAL TO THE TRANSFER PLANE				
WMAG	MAGNITUDE OF THE NON-UNITIZED WHAT VECTOR				
XF	POSITION OF THE TARGET PLANET AT DF				
XI	POSITION OF THE LAUNCH PLANET AT DI				
COMMON COMPUTED/USED:	TMU	VHPM			
COMMON COMPUTED:	DPA	NO	RAP	RIN	TIN
COMMON USED:	ALNGTH	DG	DT	IZERO	KTAR
	KUR	NLP	NTP	ONE	PI
	PMASS	RAD	SPHERE	TAR	TM
	TWO	XP	ZDAT	ZERO	

HELIO Analysis

HELIO computes the zero iterate initial state for interplanetary trajectories. The initial and final states are determined either by an arbitrary position vector or by the location of a specified planet at a prescribed time according to

- IZERO = 1 planet to planet
 2 planet to arbitrary final point
 3 arbitrary initial point to planet
 4 arbitrary initial point to final point

The final time used in locating a planet must correspond to the closest approach (CA) to the planet. Therefore if the target time is read in as a sphere of influence (SOI) time, a modification is required. The helio-centric conic is computed (as described below) using the t_{SI} time to determine the final position. The approach asymptote \vec{v}_{HP} corresponding to that trajectory is used with the desired r_{CA} to compute the time from SOI to CA. If r_{CA} is not a target variable then the target values of $B \cdot T$ and $B \cdot R$ are used to estimate the r_{CA}

$$r_{CA} = -\frac{\mu}{v_{HP}} + \frac{1}{2} \sqrt{\left(\frac{2\mu}{v_{HP}^2}\right)^2 + 4B^2} \quad (1)$$

Then the approximate approach hyperbola is given by

$$\begin{aligned} a_h &= \frac{\mu r_{SI}}{2\mu - v_{HP}^2 r_{SI}} \\ e_h &= 1 - \frac{r_{CA}}{a} \\ p_h &= a_n (1 - e_h^2) \end{aligned} \quad (2)$$

and the hyperbolic time to go from SOI to CA is given by

$$\Delta t_{SICA} = \frac{\mu}{v_{HP}^3} (e \sinh F - F) \quad (3)$$

where

$$\tanh \frac{F}{2} = \sqrt{\frac{e_n - 1}{e_h + 1}} \tanh \frac{f}{2}$$

$$\cos f = \frac{1}{e} \left(\frac{p_h}{r_{SI}} - 1 \right) \quad (4)$$

The final time is then given by $t_f = t_{SI} + \Delta t_{SICA}$.

The initial and final positions \vec{r}_i and \vec{r}_f of the heliocentric conic are either input or computed from the positions of planets determined by ORB and EPHEM. The unit normal to the heliocentric orbit plane is

$$\hat{W} = \frac{\vec{r}_i \times \vec{r}_f}{|\vec{r}_i \times \vec{r}_f|} \quad (5)$$

The inclination to that plane is

$$\cos i = \hat{W}_z \quad (6)$$

The ascending node of the plane is given by

$$\tan \Omega = \frac{\hat{W}_x}{-\hat{W}_y} \quad (7)$$

The central angle of transfer is defined by

$$\cos \Psi = \frac{\vec{r}_i \cdot \vec{r}_f}{r_i r_f} \quad (8)$$

The semi-major axis a and eccentricity e of the heliocentric conic are computed from Lambert's theorem in subroutine FLITE. The true anomaly f_i at the initial and final points are computed from

$$p = a(1 - e^2)$$

$$\cos f_i = \frac{p - r_i}{e r_i} \quad \sin f_i = \frac{\cos f_i \cos \Psi - \frac{p - r_f}{e r_f}}{\sin \Psi} \quad (8)$$

$$f_f = f_i + \Psi$$

Finally, the argument of periapsis ω is computed from

$$\cos(\omega + f_i) = \frac{\vec{r}_i \cdot \hat{U}}{r_i} \quad (10)$$

where $\hat{U} = (\cos \Omega, \sin \Omega, 0)$.

Therefore the initial or final states (\vec{r}_i, \vec{v}_i) or (\vec{r}_f, \vec{v}_f) may now be computed by ELCAR. Let (\vec{r}, \vec{v}) denote either state and let (\vec{r}_p, \vec{v}_p) denote the state of the relevant planet. The departure (or approach) asymptote is then given by

$$\vec{v}_{HP} = \vec{v}_f - \vec{v}_p \quad \vec{v}_{HE} = \vec{v}_i - \vec{v}_p \quad (11)$$

The latitude and longitude of the position vector are

$$\sin \phi = \frac{\vec{r}_y}{r} \quad \tan \theta = \frac{r_y}{r_x} \quad (12)$$

The path angle Γ may be computed from

$$\cos \Gamma = \frac{V \mu_p}{r v} \quad (13)$$

The azimuth of the relevant state is

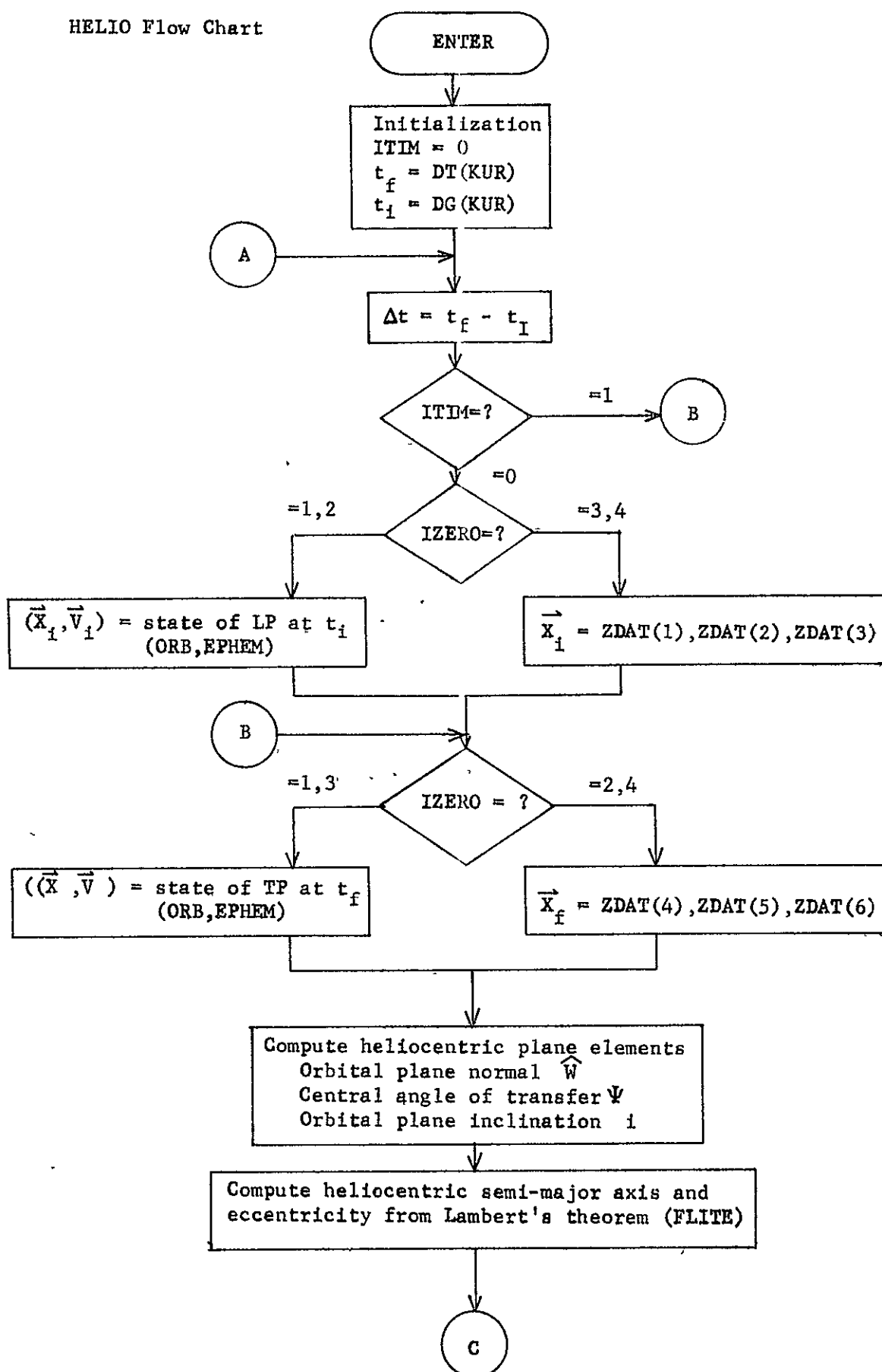
$$\sin \Sigma = \frac{(\vec{r} \times \vec{v}) \cdot \hat{U}}{|\vec{r} \times \vec{v}|} \quad (14)$$

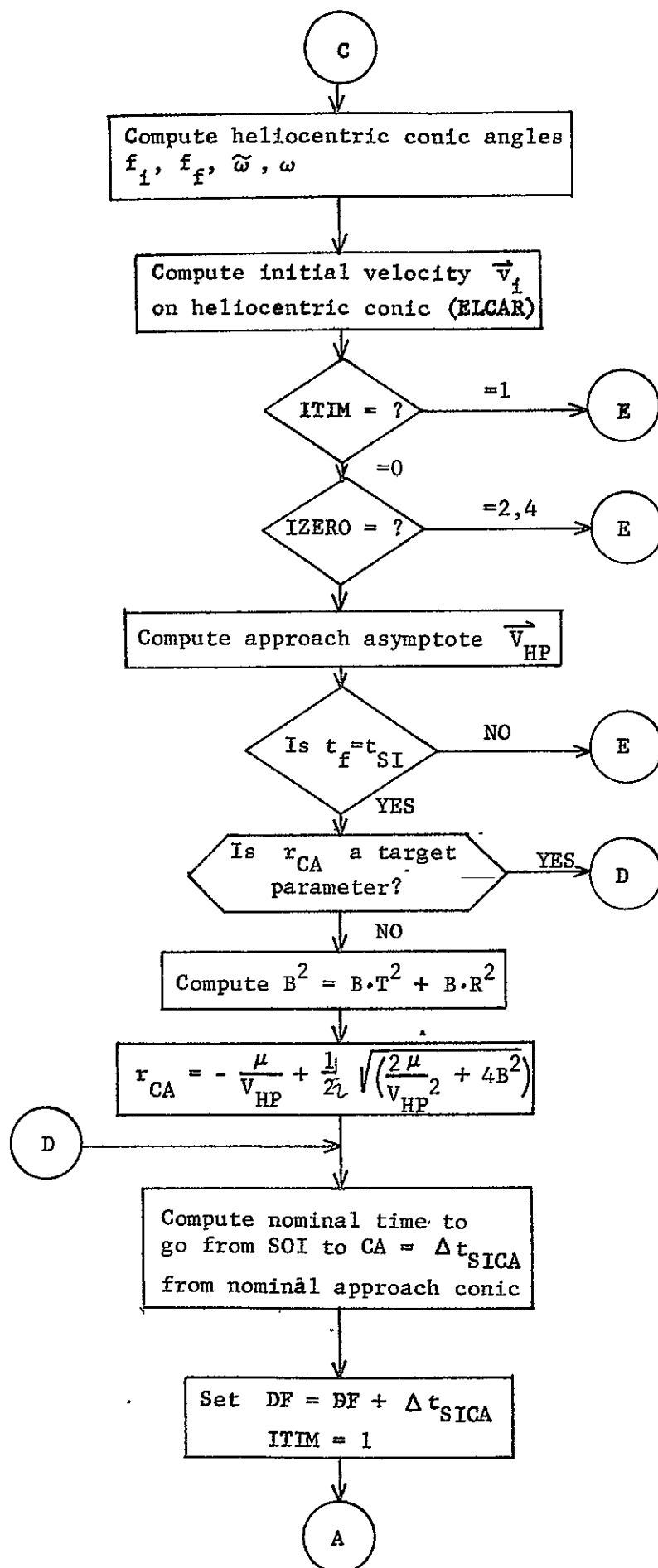
$$\cos \Sigma = \frac{\vec{v} \cdot \hat{U}}{V \cos \Gamma} \quad (15)$$

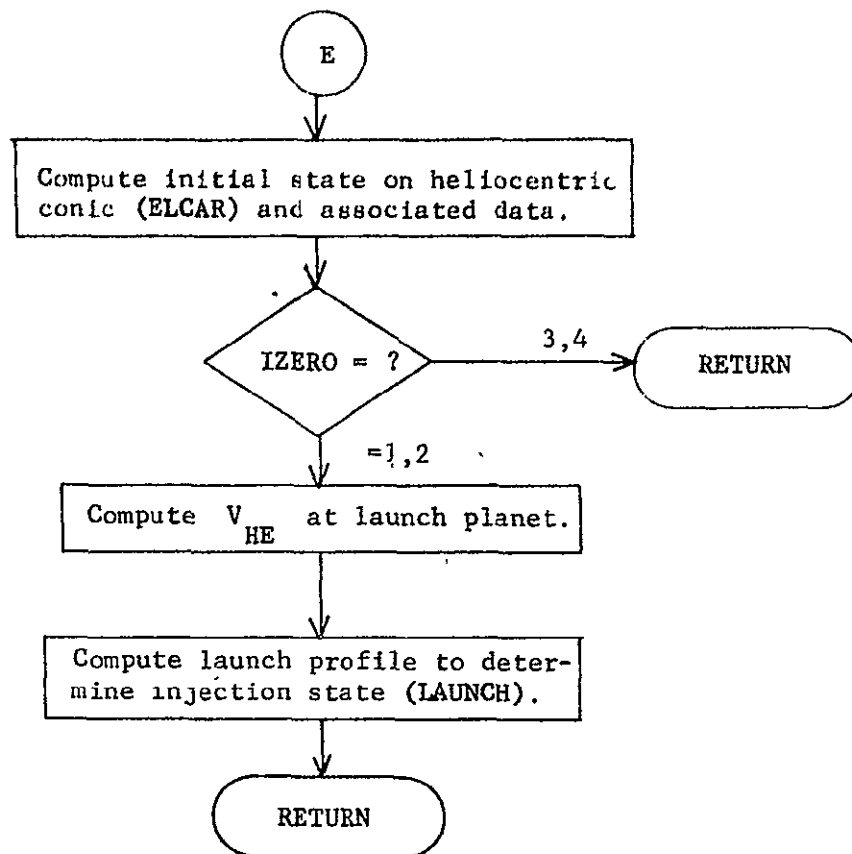
If the initial state is referenced to a planet, subroutine LAUNCH is called to convert the departure asymptote and launch profile into an injection radius, velocity, and time. Otherwise the initial state is returned as the initial state on the heliocentric conic.

Reference: Space Research Conic Program, Phase III, May 1, 1969, Jet Propulsion Laboratory, Pasadena, California.

HELIO Flow Chart







SUBROUTINE HPOST

PURPOSE: TO CALCULATE RADIUS AND TRUE ANAMOLY ON A HYPERBOLA
GIVEN TIME FROM PERIAPSIS

ARGUMENT: A I SEMI-MAJOR AXIS OF HYPERBOLA IN KM
 CTA O COSINE OF TRUE ANAMOLY
 DTPS I TIME INTERVAL FROM PERIAPSIS TO DESIRED
 STATE IN SEC
 E I ECCENTRICITY OF HYPERBOLA
 GMU I GRAVITATIONAL CONSTANT OF PLANET IN
 KM^3/SEC^2
 P I SEMI-LATUS RECTUM OF HYPERBOLA IN KM
 R O RADIUS IN KM
 STA O SINE OF TRUE ANAMOLY

SUBROUTINES SUPPORTED: TPRTRG

LOCAL SYMBOLS: ABSA ABSOLUTE VALUE OF SEMI-MAJOR AXIS IN KM
 ABSDT ABSOLUTE VALUE OF TIME FROM PERIAPSIS TO
 DESIRED STATE IN SEC
 AM1 ITERATED VALUE OF MEAN ANAMOLY
 CORRESPONDING TO CURRENT NEWTON ITERATE
 OF HYPERBOLIC ANAMOLY IN RAD
 AM MEAN ANAMOLY IN RAD
 CSF HYPERBOLIC COSINE OF HYPERBOLIC ANAMOLY
 DLE ITERATION INCREMENT TO HYPERBOLIC ANAMOLY
 DTS CONVERSION FACTOR FROM DAYS TO SECONDS
 F HYPERBOLIC ANAMOLY IN RAD
 ONE CONSTANT 1.
 P1 CONSTANT PI
 RMEAN MEAN ORBITAL RATE IN RAD/SEC
 SNF HYPERBOLIC SINE OF HYPERBOLIC ANAMOLY
 TM8 CONSTANT $1.0\text{E}-08$

TWO CONSTANT 2.

VHE MAGNITUDE OF HYPERBOLIC EXCESS VELOCITY IN
 KM/SEC

EXPONENTIAL FUNCTION OF HYPERBOLIC ANAMOLY

1-

SUBROUTINE HYELS

PURPOSE: TO COMPUTE AND PRINT THE TWO-DIMENSIONAL OR THREE-DIMENSIONAL HYPERELLIPSOID OF A SPECIFIED MATRIX.

CALLING SEQUENCE: CALL HYELS(KS,P,N)

ARGUMENT: KS I SIGMA LEVEL OF THE HYPERELLIPSOID
 P I MATRIX FOR WHICH THE HYPERELLIPSOID IS TO
 BE COMPUTED
 N I DIMENSION LIMITS OF THE SQUARE MATRIX P

SUBROUTINES SUPPORTED: EIGHTH GUISIM GUISS SETEVN GUIDM
 GUID PRED

SUBROUTINES REQUIRED: MATIN

LOCAL SYMBOLS: K2 SQUARE OF SIGMA LEVEL
 PI INVERSE OF MATRIX P
 P12 TWICE THE VALUE OF (1,2) ELEMENT OF PI
 P13 TWICE THE VALUE OF (1,3) ELEMENT OF PI
 P23 TWICE THE VALUE OF (2,3) ELEMENT OF PI
 V TEMPORARY STORAGE VECTOR FOR ARRAY P

COMMON USED: TWO

HYELS Analysis

Subroutine HYELS computes and writes out hyperellipsoids associated with a 2 or 3 dimensional covariance matrix P.

If P is assumed to be the covariance matrix of an n-dimensional random variable \hat{x} having a gaussian distribution with mean zero, then the probability density function is given by

$$p = \frac{1}{(2\pi)^{n/2} |P|^{1/2}} \exp \left[-\frac{1}{2} \hat{x}^T P^{-1} \hat{x} \right]$$

Re-writing this equation as

$$\hat{x}^T P^{-1} \hat{x} = 2 \ln \left[\frac{1}{(2\pi)^{n/2} p |P|^{1/2}} \right] = k^2$$

shows that the surface of constant probability density p is an m-dimensional ellipsoid, where m is the rank of P. The constant k can be shown to correspond to the sigma level of the ellipsoid.

For $n = 3$, the above equation has form

$$ax^2 + by^2 + cz^2 + dxy + exz + fyz = k^2$$

where	$a = a_{11}$	$d = 2a_{12}$
	$b = a_{22}$	$e = 2a_{13}$
	$c = a_{33}$	$f = 2a_{23}$

and the a_{ij} are the elements of P^{-1} .

Subroutine HYELS uses this equation to compute a 3-dimensional hyperellipsoid, and sets the appropriate constants to zero to compute a 2-dimensional hyperellipsoid.

Reference: H. Sorenson. "Kalman Filtering", Advances in Control Systems, Vol. 3, C. T. Leades (Ed.). New York: Academic Press, 1966, p. 219.

SUBROUTINE HYPT

PURPOSE: TO CALCULATE TIME FROM PERIAPSIS ON A HYPERBOLA GIVEN
TRUE ANAMOLY

ARGUMENT: CSTA I COSINE OF TRUE ANAMOLY
 EH I ECCENTRICITY OF HYPERBOLA
 ORBH I RECIPROCAL OF MEAN ORBITAL RATE IN
 SEC/RAD
 SNTA I SINE OF TRUE ANAMOLY
 T 0 TIME FROM PERIAPSIS IN SEC

SUBROUTINES SUPPORTED: CAREL IMPCT SPHIMP

LOCAL SYMBOLS: HA HYPERBOLIC ANAMOLY IN RAD
 HSNHA HYPERBOLIC SINE OF HYPERBOLIC ANAMOLY
 ONE CONSTANT 1.

SUBROUTINE IMPACT

PURPOSE: TO COMPUTE THE ACTUAL IMPACT PLANE PARAMETERS BDT AND BDR CORRESPONDING TO ANY POINT ON AN INCOMING HYPERBOLA. IT HAS THE OPTION TO CONVERT TARGET VALUES OF INCLINATION XIN AND RADIUS OF CLOSEST RCA INTO EQUIVALENT TARGET VALUES OF DBT AND DBR.

CALLING SEQUENCE: CALL IMPACT(R,V,GMX,T,BDT,BDR,XIN,RCA,DBT,DBR,TCA,KOPT)

ARGUMENTS	R(3)	I	POSITION VECTOR TO CENTRAL BODY AT EPOCH
	V(3)	I	VELOCITY VECTOR TO CENTRAL BODY AT EPOCH
	GMX	I	GRAVITATIONAL CONSTANT OF CENTRAL BODY
	T(3,3)	I	TRANSFORMATION MATRIX FROM REFERENCE TO INCLINATION SYSTEM
	BDT	O	VALUE OF ACTUAL B.T EVALUATED AT EPOCH
	BDR	O	VALUE OF ACTUAL B.R EVALUATED AT EPOCH
	XIN	I	DESIRED INCLINATION (DEG) (OPTIONAL)
	RCA	I	DESIRED RADIUS OF CLOSEST APPROACH (OPTION)
	DBT	O	TARGET VALUE OF B.T BASED ON XIN, RCA
	DBR	O	TARGET VALUE OF B.R BASED ON XIN, RCA
	TCA	O	TIME FROM PERIAPSIS ON CONIC
	KOPT	I	TARGET VALUE COMPUTATION FLAG
		=0	DO NOT COMPUTE TARGET VALUES
		=1	COMPUTE TARGET VALUES OF B.T, B.R (MUST READ IN OPTIONAL INPUT)

SUBROUTINES SUPPORTED: TAROPT LUNCON LUNTAR MULTAR VMP

SUBROUTINES REQUIRED: NONE

LOCAL SYMBOLS:	AB	INTERMEDIATE VARIABLE FOR CALCULATION OF RV,SV,TV SYSTEM
	AIN	TARGET INCLINATION IN RADIANS. AFTER NORMALIZATION
	ANG	OUTPUT VARIABLE WHEN DECLINATION CONSTRAINT IS VIOLATED

AUXF	ECCENTRIC ANOMALY (HYPERBOLIC CASE)
A	SEMI-MAJOR AXIS OF R-V CONIC
BMAG	MAGNITUDE OF DESIRED B VECTOR
BV	ACTUAL/DESIRED B VECTOR
B	MAGNITUDE OF ACTUAL B VECTOR
CDECL	COSINE OF DECL
CDELW	COSINE OF DELW
CTA	COSINE OF TA
CW	COSINE OF W
C1	MAGNITUDE OF VECTOR NORMAL TO ORBITAL PLANE IN INERTIAL SYSTEM
DB	DESIRED MAGNITUDE OF DESIRED B VECTOR
DECL	DECLINATION OF APPROACH ASYMPTOTE IN INCLINATION SYSTEM
DELW	LONGITUDE OF ASCENDING NODE IN INCLINATION SYSTEM
E	ECCENTRICITY OF THE R-V CONIC
II	INCLINATION SIGN INDICATOR. =1, INCLINATION IS POSITIVE =-1, INCLINATION IS NEGATIVE
IM	INDICATOR FOR DIRECTION OF MOTION OF THE TRAJECTORY =1, MOTION IS POSIGRADE =-1, MOTION IS RETROGRADE
PI	MATHEMATICAL CONSTANT 3.141592653589793
PV	INTERMEDIATE VECTOR USED TO CALCULATE DESIRED B VECTOR
P	SEMI-LATUS RECTUM
QV	INTERMEDIATE VECTOR USED TO CALCULATE ACTUAL B VECTOR
RAD	DEGREES TO RADIANS CONVERSION CONSTANT

RD	TIME DERIVATIVE OF RM
RM	MAGNITUDE OF THE POSITION VECTOR R
RRD	DOT PRODUCT OF R AND V VECTORS
RV	VECTOR USED TO CALCULATE ACTUAL AND DESIRED B DOT R
SDECL	SINE OF DECL
SDELW	SINE OF DELW
SINHF	HYPERBOLIC SINE OF AUXF
STA	SINE OF TA
SV	VECTOR USED TO CONSTRUCT RV, TV VECTORS. PARALLEL TO THE APPROACH ASYMPTOTE
SW	SINE OF W
SX	VARIABLE USED TO DETERMINE SIGNS OF DBT, DBR
TANG	INTERMEDIATE VARIABLE FOR CALCULATION OF AUXF
TA	TRUE ANOMALY FOR CALCULATION OF AUXF
THS	INTERMEDIATE ANGLE FOR CALCULATION OF W
TV	VECTOR USED TO CALCULATE ACTUAL AND DESIRED B DOT T
VINH	VELOCITY AT INFINITY
VX	MAGNITUDE OF THE VELOCITY VECTOR V
WMAG	MAGNITUDE OF VECTOR NORMAL TO ORBITAL PLANE IN INCLINATION SYSTEM
WV	VECTOR NORMAL TO ORBITAL PLANE IN INCLINATION AND INERTIAL SYSTEMS
W	ARGUMENT OF PERIAPSIS
Z	APPROACH ASYMPTOTE IN INCLINATION SYSTEM

COMMON USED:

NINETY ONE TWO ZERO

IMPACT Analysis

The impact parameters $B \cdot T$ and $B \cdot R$ form a convenient set of variables for the description of the approach geometry for lunar and interplanetary missions. Let a reference cartesian coordinate system XYZ (ecliptic in STEAP) be established at the center of the target body. Let \vec{V}_∞ denote the hyperbolic excess velocity of the spacecraft in the XYZ system. An auxiliary coordinate system RST may be constructed relative to the \vec{V}_∞ by the definitions

$$\begin{aligned} \hat{S} &= \vec{V}_\infty / \vec{V}_\infty & \hat{T} &= \frac{\hat{S} \times \hat{K}}{|\hat{S} \times \hat{K}|} & \hat{R} &= \hat{S} \times \hat{T} \end{aligned} \quad (1)$$

Therefore \hat{S} is in the direction of the approach asymptote, \hat{T} lies along the intersection of the impact plane (the plane normal to \hat{S} and passing through the center of the planet) and the reference plane (XY -plane), and R completes the right hand system. The \vec{B} vector lies in the impact plane and is directed to the incoming asymptote. Then $B \cdot T$ and $B \cdot R$ have the usual vector definitions.

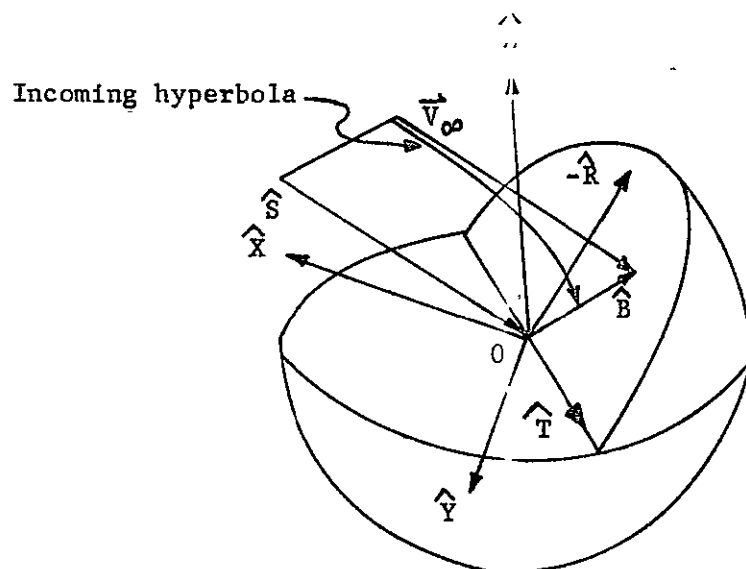


Figure 1. Impact Plane Parameters

In the optional part of the subroutine, the target impact parameter B^* associated with \hat{S} and a target inclination i (relative to target planet equator) and radius of closest approach r_{CA} is computed. However given an approach asymptote \hat{S} there are generally four trajectories with the same values of i and r_{CA} . Two of these trajectories are retrograde and

two are posigrade. For each type of motion there are two distinct planes that have the same inclination and include the \hat{S} vector. These are distinguished by the direction of motion when the approach asymptote is crossed, i.e., whether the motion is from north to south (northern approach) or from south to north (southern approach). Let $0 \leq \alpha \leq 90^\circ$. Then setting the target inclinations to the following values determines the trajectory which will be specified:

i	Trajectory
α	posigrade with northern approach
$-\alpha$	posigrade with southern approach
$180+\alpha$	retrograde with northern approach
$180-\alpha$	retrograde with southern approach

The possible trajectories are illustrated in Figure 2.

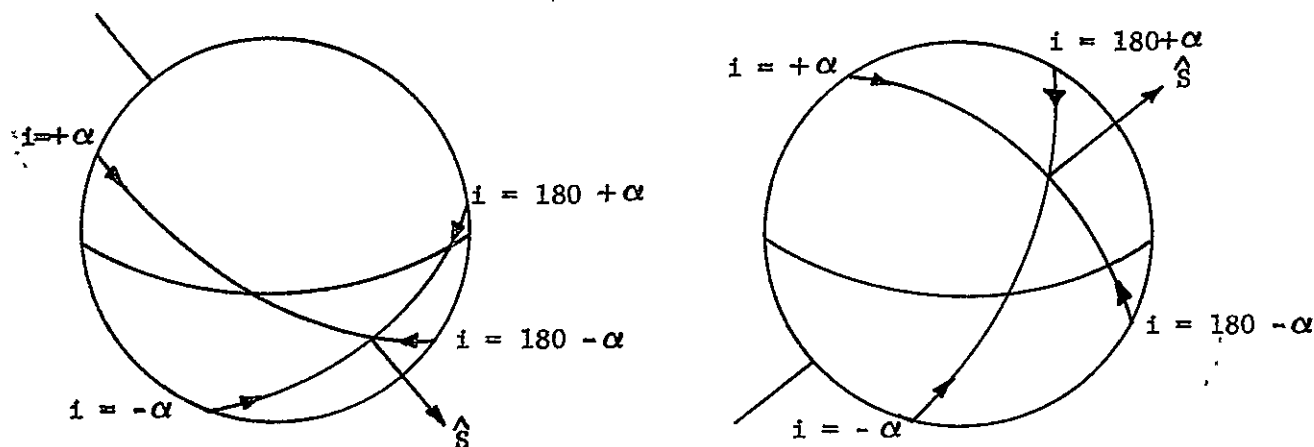


Figure 2. Possible Trajectories with Same Inclination

The detailed computations for the basic part of the program are straightforward. Using the standard conic abbreviations,

$$c = |\vec{r} \times \vec{v}| \quad (2)$$

$$\hat{w} = \frac{\vec{r} \times \vec{v}}{c} \quad (3)$$

$$p = \frac{c^2}{\mu} \quad (4)$$

$$a = \frac{r}{2 - rv^2/\mu} \quad (5)$$

$$e^2 = 1 - \frac{p}{a} \quad (6)$$

$$b = \sqrt{p |a|} \quad (7)$$

$$\cos f = \frac{p - r}{er} \quad (8)$$

$$\sin f = \frac{\dot{r}c}{e\mu} \quad (9)$$

$$\hat{Z} = \frac{r}{c} \mathbf{v} - \frac{\dot{r}}{c} \vec{r} \quad (10)$$

$$\hat{P} = \frac{\vec{r}}{r} \cos f - \hat{Z} \sin f \quad (11)$$

$$\hat{Q} = \frac{\vec{r}}{r} \sin f + \hat{Z} \cos f \quad (12)$$

$$\hat{S} = -\frac{a}{\sqrt{a^2+b^2}} \hat{P} + \frac{b}{\sqrt{a^2+b^2}} \hat{Q} \quad (13)$$

$$\hat{T} = \frac{\hat{S} \times \hat{K}}{\hat{S} \times \hat{K}} \quad (14)$$

$$\hat{R} = \hat{S} \times \hat{T} \quad (15)$$

$$\vec{B} = \frac{b^2}{\sqrt{a^2+b^2}} \hat{P} + \frac{ab}{\sqrt{a^2+b^2}} \hat{Q} \quad (16)$$

$$B \cdot T = \vec{B} \cdot \hat{T} \quad (17)$$

$$B \cdot R = \vec{B} \cdot \hat{R} \quad (18)$$

The computations for the optional part of the program which converts the i and r_{CA} into an equivalent \vec{B}^* proceed as follows. The approach asymptote is first converted into target planet equatorial coordinates and its right ascension and declination computed

$$\begin{aligned}\hat{S}_q &= \phi_{ECEQ} \hat{S} \\ \theta_S &= \tan^{-1} \frac{(S_q)_y}{(S_q)_x} \\ \delta_S &= \sin^{-1} (S_q)_z\end{aligned}\tag{19}$$

The angle $\Delta\theta$ between the ascending node of the trajectory and the right ascension of the approach asymptote is from Napier's rule

$$\sin \theta = \frac{\tan \delta_S}{\tan i}\tag{20}$$

after assuring that $|i| > |\delta_S|$. The ascending node of the trajectory is then computed recalling the definitions of the angle i

$$\Omega = \theta_S + \Delta\theta \quad (+\pi)\tag{21}$$

Thus the unit vector to the ascending node is given by

$$\hat{R}_A = (\cos \Omega, \sin \Omega, 0)\tag{22}$$

The normal to the orbital plane (in target planet equatorial coordinates) is

$$\hat{W}_q = \frac{\hat{S}_q \times \hat{R}_A}{|\hat{S}_q \times \hat{R}_A|}\tag{23}$$

This is now converted to the ecliptic coordinate system

$$\hat{W}_C = \phi_{ECEQ}^T \hat{W}_q\tag{24}$$

The unit vector in the desired \hat{B}^* direction is

$$\hat{B}^* = \frac{\hat{S} \times \hat{W}_C}{|\hat{S} \times \hat{W}_C|}\tag{25}$$

The magnitude of the \vec{B}^* vector is given by

$$B^* = r_{CA} \sqrt{1 + \frac{2\mu}{r_{CA} \dot{V}_{\infty}^2}} \quad (26)$$

Then the target impact parameter is $\vec{B}^* = B^* \hat{B}^*$. The target values are then given by their obvious definitions

$$\begin{aligned} B \cdot T^* &= \vec{B}^* \cdot \hat{T} \\ B \cdot R^* &= \vec{B}^* \cdot \hat{R} \end{aligned} \quad (27)$$

Finally the hyperbolic time from (\vec{r}, \vec{v}) to periapsis is computed from the conic formula

$$\begin{aligned} \tanh \frac{F}{2} &= \sqrt{\frac{e-1}{e+1}} \tanh \frac{f}{2} \\ t &= \sqrt{\frac{-a^3}{\mu}} (e \sinh F - F) \end{aligned} \quad (28)$$

Reference: Kizner, W., A Method of Describing Miss Distances for Lunar and Interplanetary Trajectories, Ballistic Missiles and Space Technology vol III, Pergamon Press, New York, 1961.

SUBROUTINE IMPCT

PURPOSE: TO COMPUTE FOR AUXILIARY TARGETING ACTUAL AND DESIRED B-PLANE ASYMPTOTE PIERCE POINTS AS WELL AS ACTUAL TARGET VALUES

ARGUMENT: KOPT I FLAG INDICATING SUBROUTINE OPERATING MODE
 =1 CALCULATE ACTUAL B-PLANE ASYMPTOTE
 PIERCE POINT COORDINATES (B.T AND B.R)
 ONLY
 =2 CALCULATE ACTUAL INCLINATION, RADIUS,
 AND TIME AT CLOSEST APPROACH AS WELL
 AS ACTUAL B.T AND B.R
 =3 CALCULATE ACTUAL RIGHT ASCENSION,
 DECLINATION, AND TIME AT IMPACT AS
 WELL AS ACTUAL B.T AND B.R

 R I PLANETOCENTRIC POSITION VECTOR ON WHICH
 OSCULATING CONIC IS TO BE BASED IN KM

 V I PLANETOCENTRIC VELOCITY VECTOR ON WHICH
 OSCULATING CONIC IS TO BE BASED IN KM/SEC

SUBROUTINES SUPPORTED: TAROPT

SUBROUTINES REQUIRED: HYPT MATPY SCAD USCALE UXV

LOCAL SYMBOLS: AB ECCENTRICITY TIMES MAGNITUDE OF SEMI-MAJOR
 AXIS IN KM

 AIN ACUTE INCLINATION REFERENCE ANGLE IN RAD

 A SEMI-MAJOR AXIS OF OSCULATING CONIC IN KM

 BMAG MAGNITUDE OF B VECTOR IN KM USED IN
 CALCULATION OF DESIRED B.T AND B.R FROM
 INCLINATION AND RADIUS OF CLOSEST APPROACH

 BM MAGNITUDE OF B VECTOR IN KM USED IN
 CALCULATION OF DESIRED B.T AND B.R FROM
 PROBE TARGET SITE

 BV B VECTOR ITSELF IN KM OR UNIT VECTOR IN
 DIRECTION OF B VECTOR

 CDECL COSINE OF DECLINATION OF TRAJECTORY
 ASYMPTOTE RELATIVE TO INCLINATION SYSTEM

 COELW COSINE OF DIFFERENCE BETWEEN RIGHT
 ASCENSION OF ASCENDING NODE AND RIGHT
 ASCENSION OF TRAJECTORY ASYMPTOTE RELATIVE
 TO INCLINATION SYSTEM

CSDCP	COSINE OF DECLINATION OF PROBE IMPACT SITE RELATIVE TO PROBE-SPHERE FRAME
CSDIF	COSINE OF ARC LENGTH BY WHICH DESIRED IMPACT SITE IS CLOSER TO TRAJECTORY ASYMPTOTE THAN IS PERIAPSIS
CSPHI	COSINE OF THE ANGLE BETWEEN TRAJECTORY ASYMPTOTE AND VECTOR TO DESIRED IMPACT SITE
CSRAP	COSINE OF RIGHT ASCENSION OF PROBE IMPACT SITE RELATIVE TO PROBE-SPHERE FRAME
CSTHTS	COSINE OF TRUE ANAMOLY OF TRAJECTORY ASYMPTOTE
CTA	COSINE OF TRUE ANAMOLY OF GIVEN STATE
CTPS	COSINE OF TRUE ANAMOLY AT PROBE SPHERE
CW	COSINE OF RIGHT ASCENSION OF ASCENDING NODE IN INCLINATION SYSTEM
C1	COEFFICIENT USED IN CALCULATING REPOSITIONED PROBE IMPACT SITE
C2	COEFFICIENT USED IN CALCULATING REPOSITIONED PROBE IMPACT SITE
DB	
DB	MAGNITUDE OF DESIRED B VECTOR CALCULATED FROM DESIRED INCLINATION AND RADIUS AT CLOSEST APPROACH
DECL	DECLINATION OF TRAJECTORY ASYMPTOTE IN RAD RELATIVE TO INCLINATION SYSTEM
DELW	DIFFERENCE IN RAD BETWEEN RIGHT ASCENSION OF ASCENDING NODE AND RIGHT ASCENSION OF TRAJECTORY ASYMPTOTE RELATIVE TO INCLINATION SYSTEM
DTR	CONVERSION FACTOR FROM DEGREES TO RADIANS
DTS	CONVERSION FACTOR FROM DAYS TO SECONDS
E	ECCENTRICITY OF OSCULATING CONIC
HREV	180. DEG

II	INCLINATION SIGN INDICATOR =1 INCLINATION IS POSITIVE =2 INCLINATION IS NEGATIVE
IM	INDICATOR FOR DIRECTION OF MOTION OF TRAJECTORY =1 MOTION IS POSIGRADE =2 MOTION IS RETROGRADE
ORBH	RECIPROCAL OF MEAN ORBITAL RATE IN SEC/RAD
PI	MATHEMATICAL CONSTANT PI DIRECTION OF PERIAPSIS ON OSCULATING CONIC
PV	PLANETOCENTRIC ECLIPTIC UNIT VECTOR IN
P	SEMI-LATUS RECTUM OF OSCULATING CONIC IN KM
QV	PLANETOCENTRIC ECLIPTIC UNIT VECTOR IN ORBIT PLANE ADVANCED 90 DEG FROM PV
RD	TIME RATE OF CHANGE OF OSCULATING CONIC RADIUS IN KM/SEC
RM	MAGNITUDE OF PLANETOCENTRIC OSCULATING CONIC POSITION VECTOR IN KM
RPR	RADIUS OF PROBE SPHERE IN KM
RPVRV	PLANETOCENTRIC ECLIPTIC UNIT VECTOR TO IMPACT SITE
RPV	PLANETOCENTRIC PROBE-SPHERE UNIT VECTOR TO IMPACT SITE
RRD	PRODUCT OF OSCULATING CONIC RADIUS BY ITS TIME RATE OF CHANGE IN KM^2/SEC
RTD	CONVERSION FACTOR FROM RADIANS TO DEGREES
RV	PLANETOCENTRIC ECLIPTIC UNIT VECTOR IN DIRECTION OF CROSS PRODUCT OF OSCULATING CONIC ASYMPTOTE BY ECLIPTIC POLE VECTOR
SDECL	SINE OF DECLINATION OF OSCULATING CONIC ASYMPTOTE RELATIVE TO INCLINATION SYSTEM
SDELW	SINE OF DIFFERENCE BETWEEN RIGHT ASCENSION OF ASCENDING NODE AND RIGHT ASCENSION OF ASYMPTOTE FOR OSCULATING CONIC RELATIVE TO INCLINATION SYSTEM

SNDCP	SINE OF DECLINATION PROBE IMPACT SITE RELATIVE TO PROBE-SPHERE FRAME
SNPHI	SINE OF ANGLE BETWEEN OSCULATING CONIC ASYMPTOTE AND VECTOR TO DESIRED IMPACT SITE
SNRAP	SINE OF RIGHT ASCENSION OF PROBE IMPACT SITE RELATIVE TO PROBE-SPHERE FRAME
SNTHTS	SINE OF TRUE ANAMOLY OF OSCULATING CONIC ASYMPTOTE
STA	SINE OF TRUE ANAMOLY ON OSCULATING CONIC AT GIVEN STATE
STPS	SINE OF TRUE ANAMOLY ON OSCULATING CONIC AT PROBE SPHERE
SV	PLANETOCENTRIC ECLIPTIC UNIT VECTOR IN DIRECTION OF OSCULATING CONIC ASYMPTOTE
SW	SINE OF RIGHT ASCENSION OF ASCENDING NODE OF OSCULATING CONIC IN INCLINATION SYSTEM
TEST	QUANTITY USED TO DETERMINE PROPER POLARITY OF WV
THS	RIGHT ASCENSION IN DEG OF OSCULATING CONIC ASYMPTOTE REALTIVE TO INCLINATION SYSTEM
TPSCA	TIME INTERVAL IN SEC FROM PROBE SPHERE TO CLOSEST APPROACH ON OSCULATING CONIC
TSICA	TIME INTERVAL IN SEC FROM SOI TO CLOSEST APPROACH
TV	CROSS PRODUCT OF SV BY RV
VINF	MAGNITUDE OF HYPERBOLIC EXCESS VELOCITY IN KM/SEC
VS	SQUARE OF MAGNITUDE OF PLANETOCENTRIC VELOCITY IN KM^2/SEC^2
WEQ	PLANETOCENTRIC EQUATORIAL UNIT VECTOR IN DIRECTION OF OSCULATING CONIC ANGULAR MOMENTUM
WMAG	UN-NORMALIZED MAGNITUDE OF WV

WV PLANETOCENTRIC ECLIPTIC VECTOR IN
 DIRECTION OF OSCULATING CONIC ANGULAR
 MOMENTUM

W RIGTH ASCENSION OF ASCENDING NODE OF
 OSCULATING CONIC RELATIVE TO INCLINATION
 SYSTEM

Z PLANETOCENTRIC EQUATORIAL UNIT VECTOR IN
 DIRECTION OF OSCULATING CONIC ASYMPTOTE
 OR PLANETOCENTRIC ECLIPTIC UNIT VECTOR IN
 ORBIT PLANE 90 DEG ADVANCED FROM GIVEN
 POSITION VECTOR

COMMON COMPUTED/USED: B DCP DIN RAP XATAR

COMMON COMPUTED: AAUX BDR BDT DBR DBT

COMMON USED: CAINC DC DDCP DINC DRAP
 DRCA DSI DTAR ECSS EQECP
 FOUR GMX HALF IINGRA IPHASE
 IRCADC ITARR KEYTAR KUR NINETY
 NOPAR ONE RCA RPS TWO
 ZERO

IMPCT Analysis

The subroutine IMPCT is responsible for computing all of the target parameter data associated with auxiliary targeting. Three basic types of target information are required given a planetocentric ecliptic state. First, what are the actual B-plane pierce point coordinates, $B_A \cdot T$ and $B_A \cdot R$? Second, what are the values of the actual target parameters? These may be triples of inclination, radius and time at closest approach or right ascension, declination, and time at impact. Third, what are the B-plane pierce point coordinates, $B_D \cdot T$ and $B_D \cdot R$ on the current trajectory required to achieve the desired values of the corresponding actual target parameters. In addition to supplying all of this information, IMPCT places it in the appropriate locations for sorting by the processing routine TAROPT.

Whenever IMPCT is called, it first calculates the actual B-plane pierce point coordinates for the current state. In the process it also calculates other useful information about the osculating conic, including the parameters a , e , θ , W , S , T and R . For the equations giving these quantities see the subroutine STIMP analysis. If option flag KOPT is 1, only this information is desired and a return to the calling program is executed.

The values of the actual target parameters are calculated if KOPT is not 1. If the targets are inclination, radius, and time at closest approach, KOPT must be 2. If TARGET is not in the second phase of a two-phase targeting case, the target parameters, i , r_{CA} , and t_{CA} are obtained by conic extrapolation from the current state (SOI):

$$r_{CA} = a(1 - e). \quad (1)$$

Let D denote the transformation from planetocentric ecliptic coordinates to the planetocentric equatorial frame:

$$i = \cos^{-1} \left[(D W)_3 \right] \quad (2)$$

If TARGET is in a second phase, the virtual mass program will have integrated the trajectory all the way to closest approach rather than stopping at the SOI. Hence refined values of all three target parameters are available from VMP.

Suppose, on the other hand, that the targets are right ascension α , declination δ , and time t_I at impact. Then KØPT must be 3.

Again if TARGET is not involved in a second phase, these target parameters are calculated by extrapolating conically from the SOI. Let \underline{r}_I and $\underline{\rho}_I$ denote the position vectors of the vehicle at impact in the planetocentric ecliptic and probe-sphere frames, respectively. Let C denote the transformation from the former frame to the latter. Obviously

$$\underline{\rho}_I = C \underline{r}_I. \quad (4)$$

The right ascension and declination at impact are then

$$\alpha = \tan^{-1} \left[\left(\rho_I \right)_2 / \left(\rho_I \right)_1 \right], \quad (5)$$

and

$$\delta = \sin^{-1} \left[\left(\rho_I \right)_3 / \rho_I \right]. \quad (6)$$

Let Δt_{SC} and Δt_{IC} be the times from the sphere of influence and from the probe sphere to closest approach, respectively. Let θ_I denote the true anomaly on the osculating conic at the probe sphere:

$$\cos \theta_I = (p / r_I - 1/a) / e \quad (7)$$

where

$$p = a (1 - e^2) \quad (8)$$

is the semilatus rectum of the conic. Since impact occurs before peripsis,

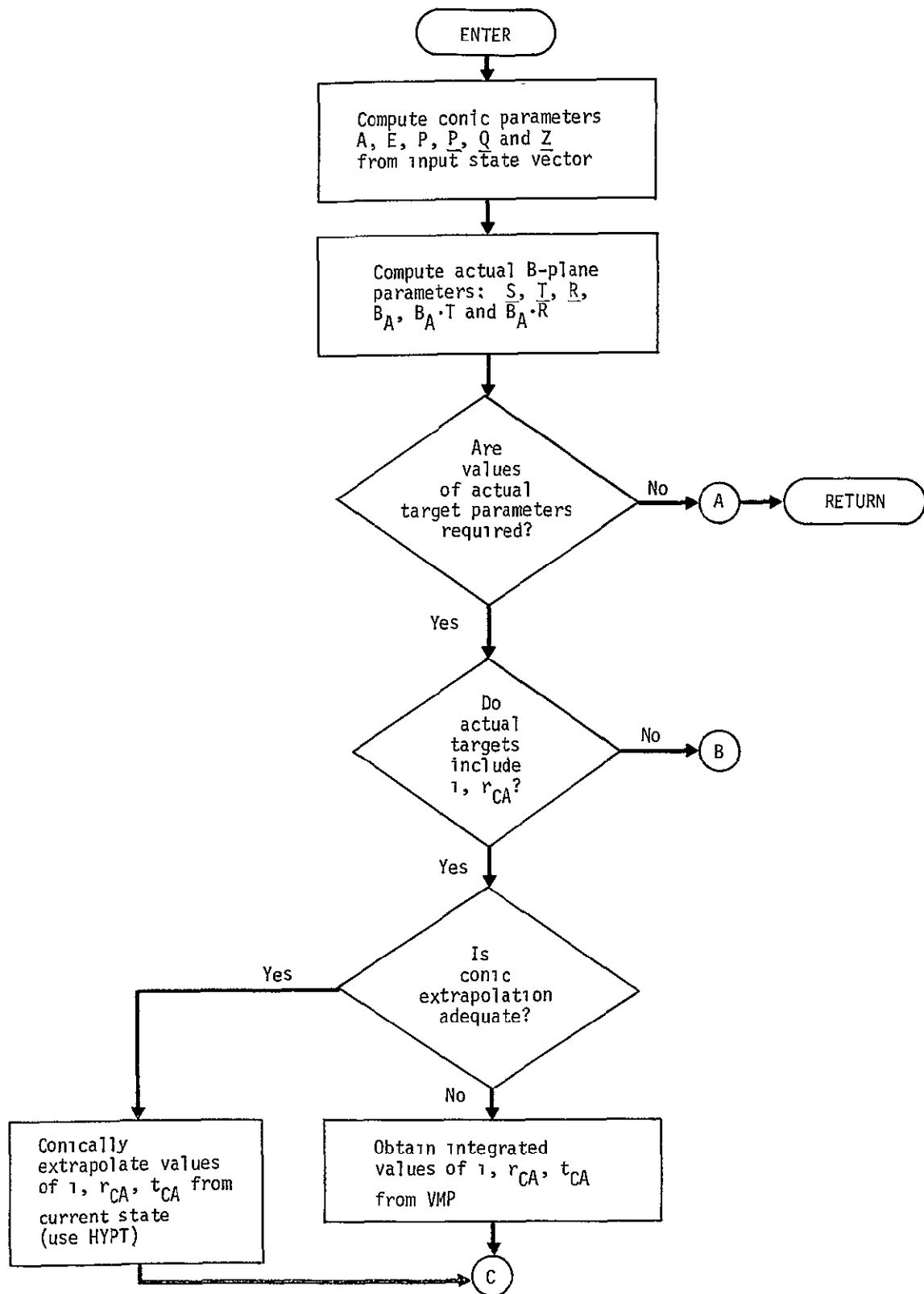
$$\sin \theta_I = - \sqrt{1 - \cos^2 \theta_I}. \quad (9)$$

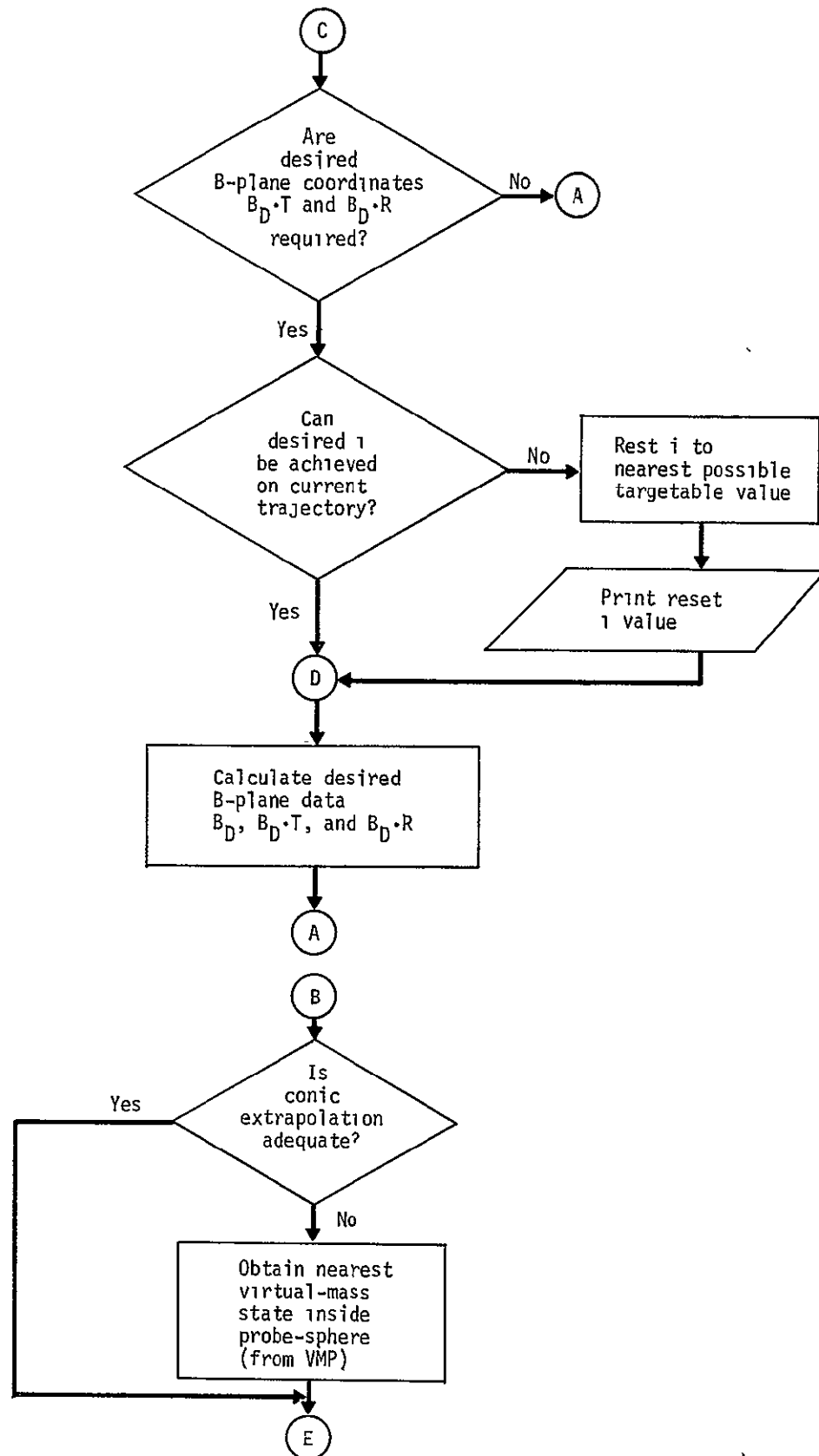
With the true anomalies on the conic at the current state and at the probe sphere available, IMPCT calls HYPT to determine Δt_{SC} and Δt_{IC} . If, as above, t_{SOI} denotes the time from the sphere of influence to periapsis,

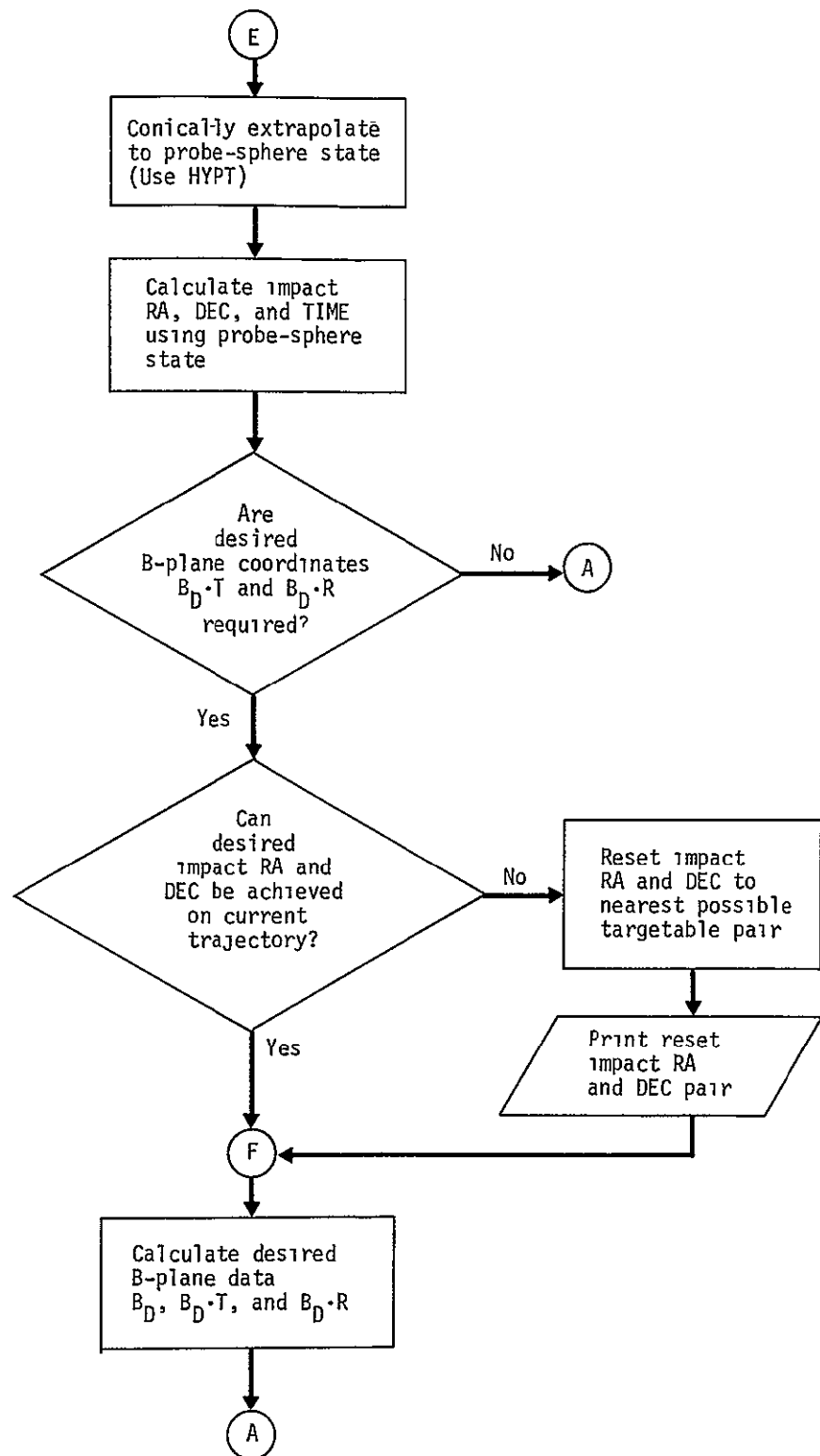
$$t_I = t_{SOI} + (\Delta t_{SC} - \Delta t_{IC}). \quad (10)$$

Finally, if TARGET is involved in the second phase of targeting, the virtual-mass algorithm integrates the trajectory all the way to the first integration increment inside the probe sphere. Hence the preceding conic extrapolation formulae can be used to obtain accurate impact target parameters by replacing the state at the SOI with the first state inside the probe sphere.

IMPCT calculates the desired B-plane pierce point coordinates $B_D \cdot T$ and $B_D \cdot R$ for either the i and r_{CA} or α and β target options if the flag ITARR is 2 indicating that a new control iteration is being made. The equations and logical flow of this calculation for the former target option is given in the subroutine IMPACT while those for the latter are presented in DIMPCP.







SUBROUTINE INPUTZ

PURPOSE: TO CONVERT THE INPUT INFORMATION FOR THE VIRTUAL MASS PROGRAM INTO VARIABLES COMPATIBLE WITH THE REST OF THE VIRTUAL MASS SUBROUTINES

CALLING SEQUENCE: CALL INPUTZ(RS,NTP,IPRINT)

ARGUMENTS	RS(6)	I	INERTIAL STATE OF S/C AT INITIAL TIME
	NTP	I	CODE OF TARGET BODY
	IPRINT	I	INITIAL INFORMATION PRINT FLAG
			=0 PRINT INITIAL DATA
			=1 DO NOT PRINT INITIAL DATA

SUBROUTINES SUPPORTED: VMP

SUBROUTINES REQUIRED: TIME SPACE

LOCAL SYMBOLS:	D	INTERMEDIATE VARIABLE FOR PRINTOUT PURPOSES
	D2	JULIAN DATE OF FINAL TRAJECTORY TIME
	IDAY	DAY OF CALENDAR DATE OF FINAL TRAJECTORY TIME
	IHR	HOUR OF CALENDAR DATE OF FINAL TRAJECTORY TIME
	IMIN	MINUTE OF CALENDAR DATE OF FINAL TRAJECTORY TIME
	IMO	MONTH OF CALENDAR DATE OF FINAL TRAJECTORY TIME
	INERR	NOT USED
	IP	CODE OF I-TH PLANET FOR STORAGE OF PMASS ARRAY
	IYR	YEAR OF CALENDAR DATE OF FINAL TRAJECTORY TIME
	LDAY	DAY OF CALENDAR DATE OF INITIAL TIME
	LHR	HOUR OF CALENDAR DATE OF INITIAL TIME
	LMIN	MINUTE OF CALENDAR DATE OF INITIAL TIME
	LMO	MONTH OF CALENDAR DATE OF INITIAL TIME

LYR YEAR OF CALENDAR DATE OF INITIAL TIME
 SECI SECOND OF CALENDAR DATE OF FINAL TIME
 SECL SECOND OF CALENDAR DATE OF INITIAL TIME
 TP INTERMEDIATE VARIABLE FOR CALCULATION OF
 COMPUTING INTERVAL

COMMON COMPUTED/USED: V

COMMON COMPUTED: F INC IPR ITRAT KOUNT
 NBODY

COMMON USED: NBOYI NO PMASS ZERO

SUBROUTINE INSERTS

PURPOSE: TO CONTROL THE PROCESSING OF AN ORBITAL INSERTION EVENT.

CALLING SEQUENCE: CALL INSERS(DTIME)

ARGUMENT: DTIME 0 TIME INTERVAL FROM DECISION TO EXECUTION (DAYS)

SUBROUTINES SUPPORTED: GIDANS

SUBROUTINES REQUIRED: COPINS NONINS PECEQ

LOCAL SYMBOLS:	DA	DESIRED SEMIMAJOR AXIS
	DE	DESIRED ECCENTRICITY
	DI	DESIRED INCLINATION
	DN	DESIRED LONGITUDE OF ASCENDING NODE
	DWTP	DESIRED ARGUMENT OF PERIAPSIS SHIFT OR DESIRED ARGUMENT OF PERIAPSIS
	ECEQI	ECLIPTIC TO EQUATORIAL TRANSFORMATION
	GM	GRAVITATIONAL CONSTANT OF TARGET BODY
	IEX	UNEXECUTABLE EVENT CODE =0 EVENT IS EXECUTABLE =1 NO EXECUTABLE SOLUTION FOUND
	IOPT	INSERTION STRATEGY OPTION =1 COPLANAR INSERTION =2 NONPLANER INSERTION
	RSP	SPACECRAFT POSITION IN ECLIPTIC COORDS
	RSQ	SPACECRAFT POSITION IN EQUATORIAL COORDS
	TEX	TIME INTERVAL TO EXECUTION (SECONDS)
	VSP	SPACECRAFT VELOCITY IN ECLIPTIC COORDS
	VSQ	SPACECRAFT VELOCITY IN EQUATORIAL COORDS

COMMON COMPUTED/USED: DELTAV

COMMON COMPUTED: DELV KTIM KWT

COMMON USED:	ALNGTH	D1	F	KMXQ	KTAR
	KUR	NBOD	NB	NTP	PHASS

INSERS Analysis

INSERS controls the processing of an orbital insertion event. The subroutine COPINS and NONINS perform the actual computations for the coplanar and non-planar options respectively.

INSERS first records the specific parameter values for the current orbit insertion event.

It then computes the current state (\vec{r} , \vec{v}) of the spacecraft in target-planet centered ecliptic coordinates. Subroutine PECEQ is called to compute the transformation matrix Φ_{ECEQ} from ecliptic to equatorial coordinates. The planet centered equatorial coordinates are then

$$\begin{aligned}\vec{r}_q &= \Phi_{ECEQ} \vec{r} \\ \vec{v}_q &= \Phi_{ECEQ} \vec{v}\end{aligned}$$

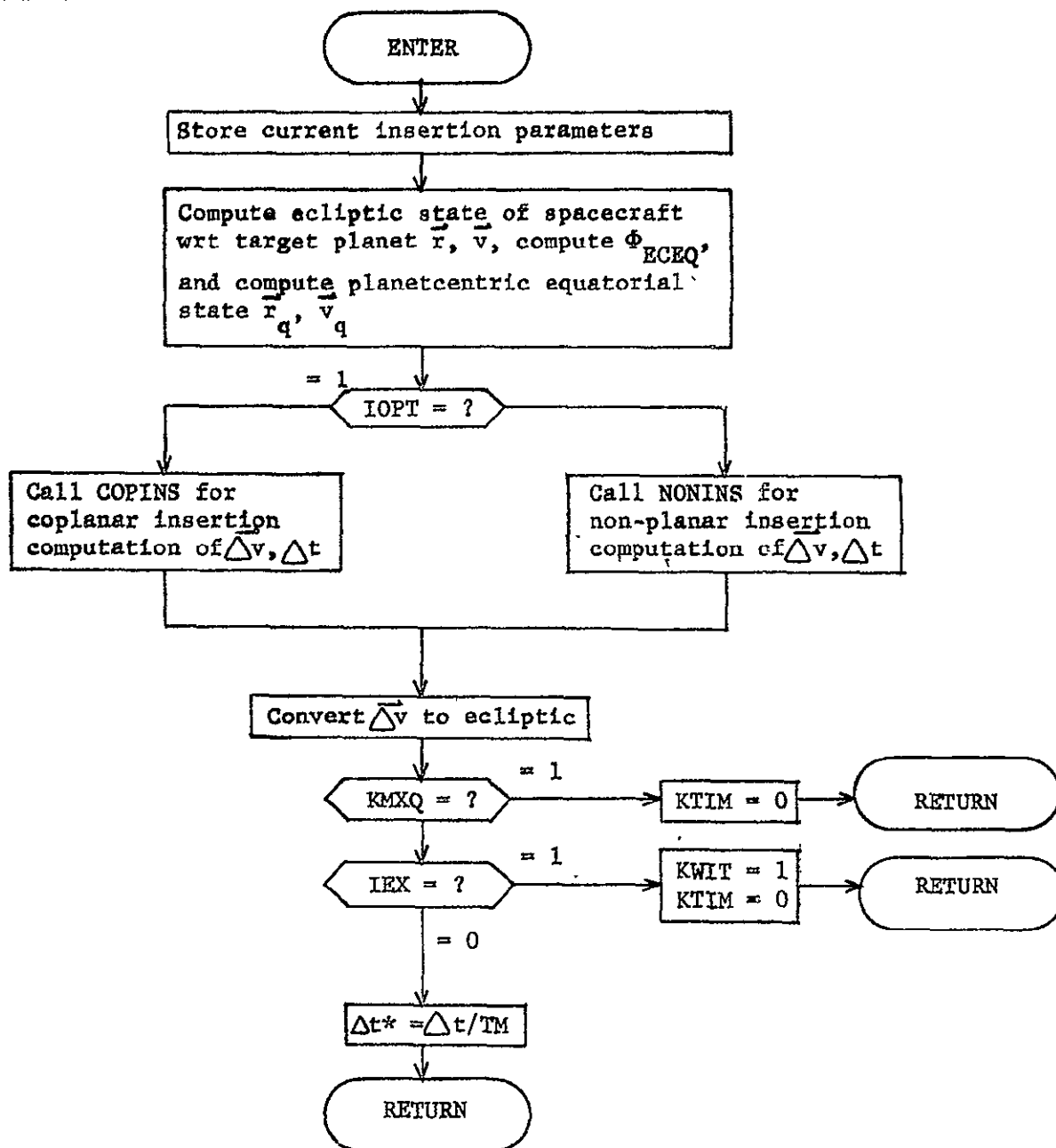
This state is then sent to COPINS or NONINS for the computation of the insertion velocity $\Delta \vec{v}$ and the time interval t between the current time and the time at which the insertion should take place (based on conic propagation about the target body). The correction $\Delta \vec{v}_q$ is then converted to ecliptic coordinates

$$\Delta \vec{v} = \Phi^T \Delta \vec{v}_q$$

If the event is a compute-only mode, the return is made to GIDANS.

If the event is to be executed the flag IEX (set by COPINS or NONINS to indicate success or failure) is then interrogated. If IEX = 1, no acceptable insertion event was found and so the executive flag KWT is set to 1 before returning. If IEX = 0 an acceptable insertion was determined and so it is set up.

INSERS Flow Chart



SUBROUTINE JACOB

PURPOSE: TO APPROXIMATE BY DIVIDED DIFFERENCES THE JACOBIAN SENSITIVITY MATRIX OF A VECTOR-VALUED FUNCTION WITH RESPECT TO A VECTOR ARGUMENT

ARGUMENT:	DELTA	I	COMMON PERTURBATION SIZE USED FOR ALL COMPONENTS OF INDEPENDENT VECTOR
	FPHI	I	FUNCTION RELATING DEPENDENT VECTOR TO INDEPENDENT VECTOR
	M	I	DIMENSION OF DEPENDENT VECTOR
	N	I	DIMENSION OF INDEPENDENT VECTOR
	PHI	I	DEPENDENT VECTOR
	RJCBM	O	JACOBIAN SENSITIVITY MATRIX OF PHI WITH RESPECT TO X
	X	I	INDEPENDENT VECTOR

SUBROUTINES SUPPORTED: GAUSLS

SUBROUTINES REQUIRED: FPHI

LOCAL SYMBOLS: IZERO

PHIN PERTURBED VALUES OF DEPENDENT VECTOR

XSAVE ORIGINAL VALUE OF COMPONENT OF INDEPENDENT VECTOR CURRENTLY BEING PERTURBED

SUBROUTINE JACOBI

PURPOSE: TRANSFORMATION OF A REAL SYMMETRIC MATRIX TO DIAGONAL FORM BY A SUCCESSION OF PLANE ROTATIONS TO ANNIHILATE THE OFF-DIAGONAL ELEMENTS AND SUBSEQUENT COMPUTATION OF THE EIGENVALUES AND EIGENVECTORS OF THAT MATRIX

CALLING SEQUENCE: CALL JACOBI(A,W2,V,N,FOD)

ARGUMENT: A I MATRIX TO BE DIAGONALIZED (WILL BE DESTROYED)

W2 O VECTOR OF EIGENVALUES (LENGTH N)

V O MATRIX OF EIGENVECTORS (N BY N DIMENSION)

N I DIMENSION OF SQUARE MATRIX A

FOD I FINAL OFF-DIAGONAL ANNIHILATION VALUE

SUBROUTINES SUPPORTED: EIGHY GUISIM GUISS PRESIM SETEVN
GUIDM GUID PRED

LOCAL SYMBOLS: AIIP INTERMEDIATE VARIABLE

AIPIP INTERMEDIATE VARIABLE-A(IPIP)

AIPJP INTERMEDIATE VARIABLE-A(IPJP)

AJPJP INTERMEDIATE VARIABLE-A(JPJP)

CS INTERMEDIATE VARIABLE

DEL DIFFERENCE IN ELEMENTS OF A

IREDO COUNTER

KR DIMENSION OF A

KRP1 KR + 1

NM1 N - 1

RAD INTERMEDIATE VARIABLE

SN INTERMEDIATE VARIABLE

TN INTERMEDIATE VARIABLE

Y1 LARGEST OFF-DIAGONAL ELEMENT

VIIP INTERMEDIATE VARIABLE

JACOBI Analysis

The Jacobi method subjects a real, symmetric matrix A to a sequence of transformations based on a rotation matrix:

$$O_K = \begin{bmatrix} \cos \phi_K & -\sin \phi_K \\ \sin \phi_K & \cos \phi_K \end{bmatrix}$$

where all other elements of the rotation matrix are identical with the unit matrix. After n multiplications A is transformed into:

$$A' = O_N^{-1} \dots O_1^{-1} A O_1 \dots O_N$$

If ϕ_K is chosen at each step to make a pair of off-diagonal elements zero, then A' will approach diagonal form with the eigenvalues on the diagonal. The columns of $O_1 O_2 \dots O_N$ correspond to the eigenvectors of A .

The angle of rotation ϕ is chosen in the following way. If the four entries of O_K are in (i,i) , (i,j) , (j,i) and (j,j) then the corresponding elements of $O_1^{-1} A O_1$ are

$$\begin{aligned} b_{ii} &= a_{ii} \cos^2 \phi + 2a_{ij} \sin \phi \cos \phi + a_{jj} \sin^2 \phi \\ b_{ij} &= b_{ji} = (a_{jj} - a_{ii}) \sin \phi \cos \phi + a_{ij} (\cos^2 \phi - \sin^2 \phi) \\ b_{jj} &= a_{ii} \sin^2 \phi - 2a_{ij} \sin \phi \cos \phi + a_{jj} \cos^2 \phi \end{aligned}$$

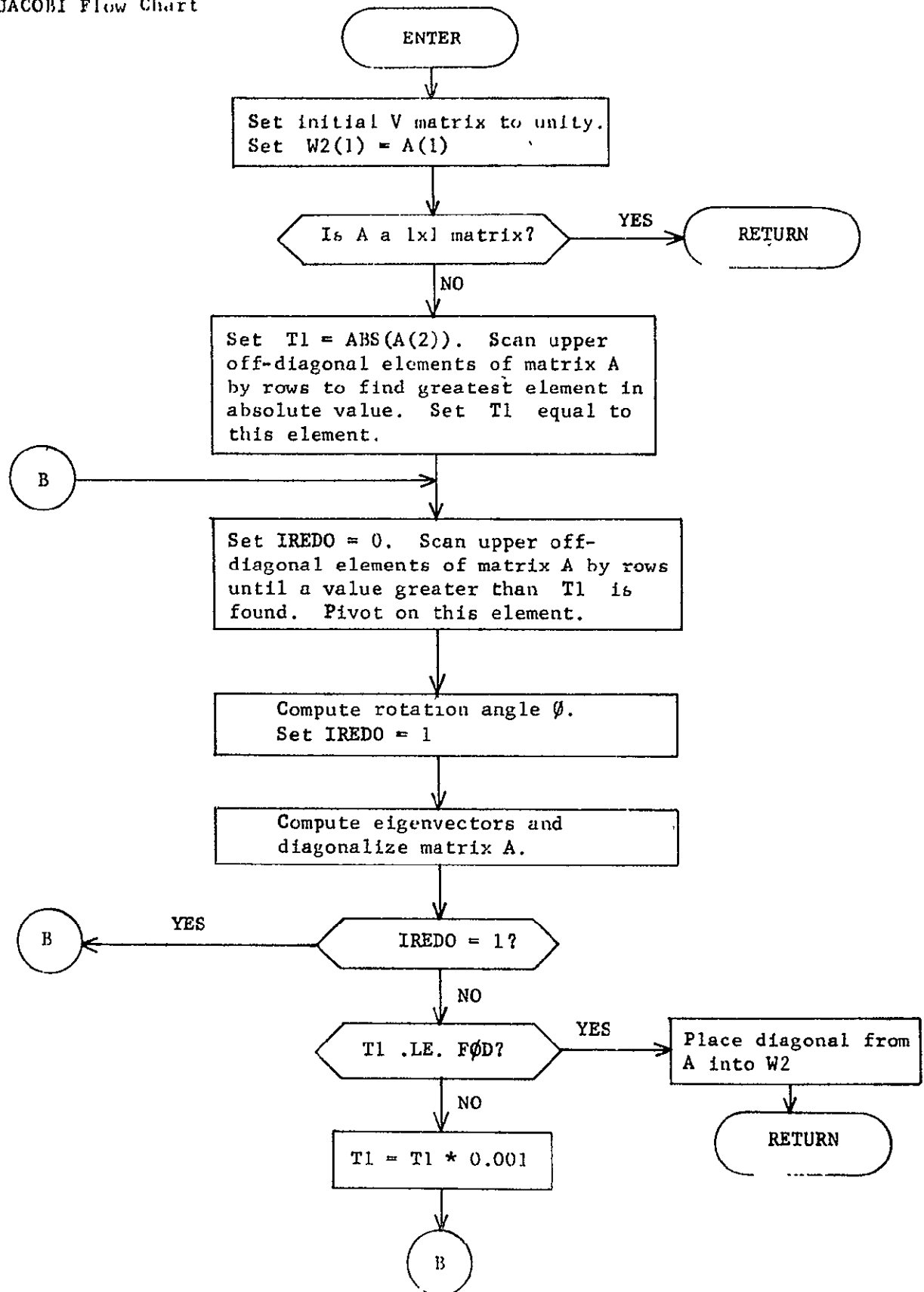
If ϕ is chosen so that $\tan 2\phi = 2a_{ij}/(a_{ii} - a_{jj})$ then

$$b_{ij} = b_{ji} = 0$$

Each multiplication creates a new pair of zeros but will introduce a non-zero contribution to positions zeroed out on previous steps. However, successive matrices of the form $O_2^{-1} O_1^{-1} A O_1 O_2$ will approach the required diagonal form.

Reference: Scheid, Frances: Theory and Problems of Numerical Analysis, McGraw-Hill Book Company, Inc., New York, 1968.

JACOBI Flow Chart



SUBROUTINE KTROL

PURPOSE: TO COMPUTE HELIOCENTRIC ECLIPTIC VELOCITY CORRECTIONS
GIVEN LAUNCH-PLANETOCENTRIC VELOCITY CONTROLS

AUGUMENT: CON I VECTOR OF VELOCITY CONTROLS IN KM/SEC AND
RAD

DV O CORRECTION IN HELIOCENTRIC VELOCITY IN
KM/SEC PRODUCED BY VELOCITY CONTROLS

IOPT I FLAG INDICATING WHICH CONTROLS ARE TO BE
PERTURBED

X I LAUNCH-PLANETOCENTRIC ECLIPTIC STATE
VECTOR AT MANEUVER POINT IN KM AND KM/SEC

SUBROUTINES SUPPORTED: TARMAX TARGET

SUBROUTINES REQUIRED: USCALE UXV

LOCAL SYMBOLS: CSC2 COSINE OF IN-PLANE ROTATION ANGLE

CSC3 COSINE OF OUT-OF-PLANE ROTATION ANGLE

ONE CONSTANT 1.

RTD CONVERSION FACTOR FROM RADIANS TO DEGREES

SNC2 SINE OF IN-PLANE ROTATION ANGLE

SNC3 SINE OF OUT-OF-PLANE ROTATION ANGLE

U CROSS PRODUCT OF W BY V

V LAUNCH-PLANETOCENTRIC ECLIPTIC UNIT VECTOR
IN DIRECTION OF VELOCITY AT MANEUVER
POINT

VM MAGNITUDE OF VELOCITY VECTOR AT MANEUVER
POINT

W LAUNCH-PLANETOCENTRIC ECLIPTIC UNIT VECTOR
IN DIRECTION OF ANGULAR MOMENTUM AT
MANEUVER POINT

WM MAGNITUDE OF CROSS PRODUCT OF LAUNCH-
PLANETOCENTRIC ECLIPTIC POSITION VECTOR
AND UNIT VELOCITY VECTOR AT MANEUVER POINT

KTRØL Analysis

KTRØL calculates the targeting velocity increment $\underline{\Delta v}$ given the targeting state vector $\underline{X} = (\underline{r}/\underline{v})^T$ of the spacecraft relative to the launch planet and the launch-planetocentric velocity controls c_1 , c_2 , and c_3 . This computation is required in two distinct situations. The first is in calculating the sensitivity matrix of the auxiliary parameters to the velocity controls by successively perturbing each control while holding the remaining two constant. The second is in applying the control correction indicated by the Newton-Raphson algorithm to arrive at the next iterate to the postimpulse targeting state.

In either case the three unit vectors \underline{V} , \underline{U} , and \underline{W} that serve to define the local, spherical, velocity-control coordinate system are first computed

$$\underline{V} = \frac{\underline{v}}{v} \quad (1)$$

$$\underline{W} = \frac{\underline{r} \times \underline{V}}{\|\underline{r} \times \underline{V}\|} \quad (2)$$

$$\underline{U} = \underline{W} \times \underline{V} . \quad (3)$$

\underline{V} specifies the direction of zero latitude and zero longitude in the control frame while the \underline{W} axis determines the +z or polar direction. Then c_2 and c_3 are, respectively, the latitude and longitude of the posttargeting velocity while c_1 is the increase in length of that velocity. Figure 1 defines the controls pictorially when the earth is the launch planet. The velocity increments required in either of the two situations mentioned above can readily be calculated in terms of the vector \underline{V} , \underline{U} , and \underline{W} .

First consider the calculation of the increment $\underline{\Delta v}_i$ produced by perturbing the i th control an amount c_i , while fixing the other controls at zero as required in the sensitivity approximation. KTRØL performs this computation when IOPT = i. Reasoning from Figure 1 it follows immediately that

$$\underline{\Delta v}_1 = c_1 \underline{V} \quad (4)$$

$$\underline{\Delta v}_2 = \|\underline{v}\| \left(\cos c_2 - 1 \right) \underline{V} + \sin c_2 \underline{U} \quad (5)$$

$$\underline{\Delta v}_3 = \|\underline{v}\| \left(\cos c_3 - 1 \right) \underline{V} + \sin c_3 \underline{W} . \quad (6)$$

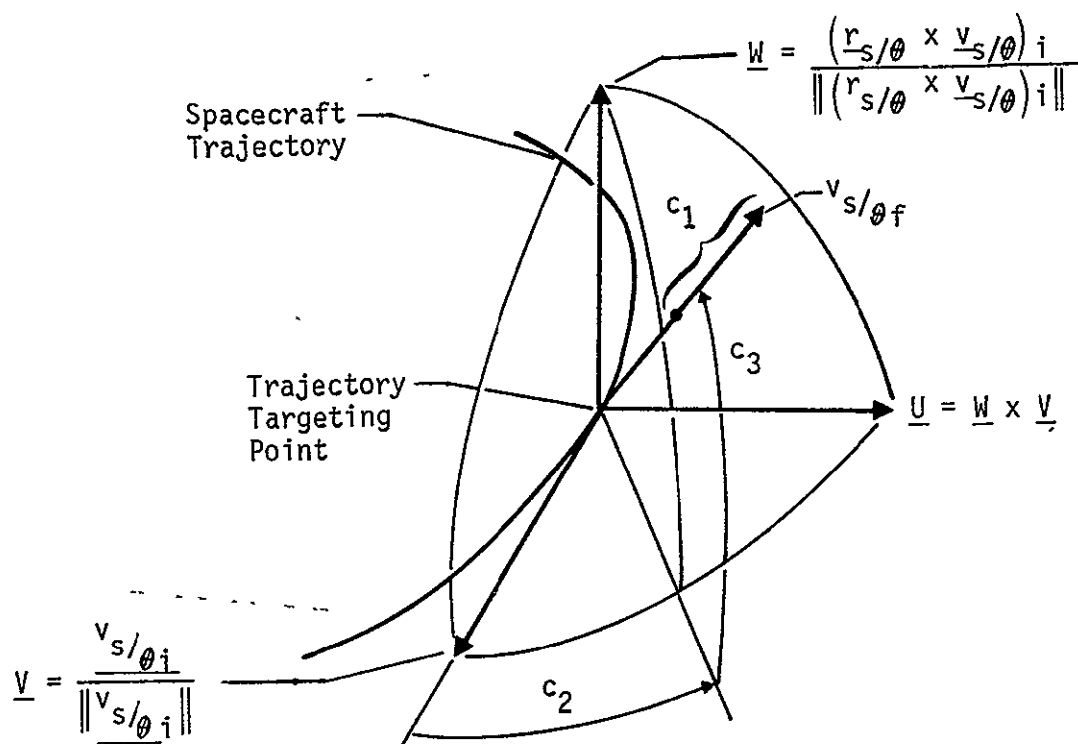


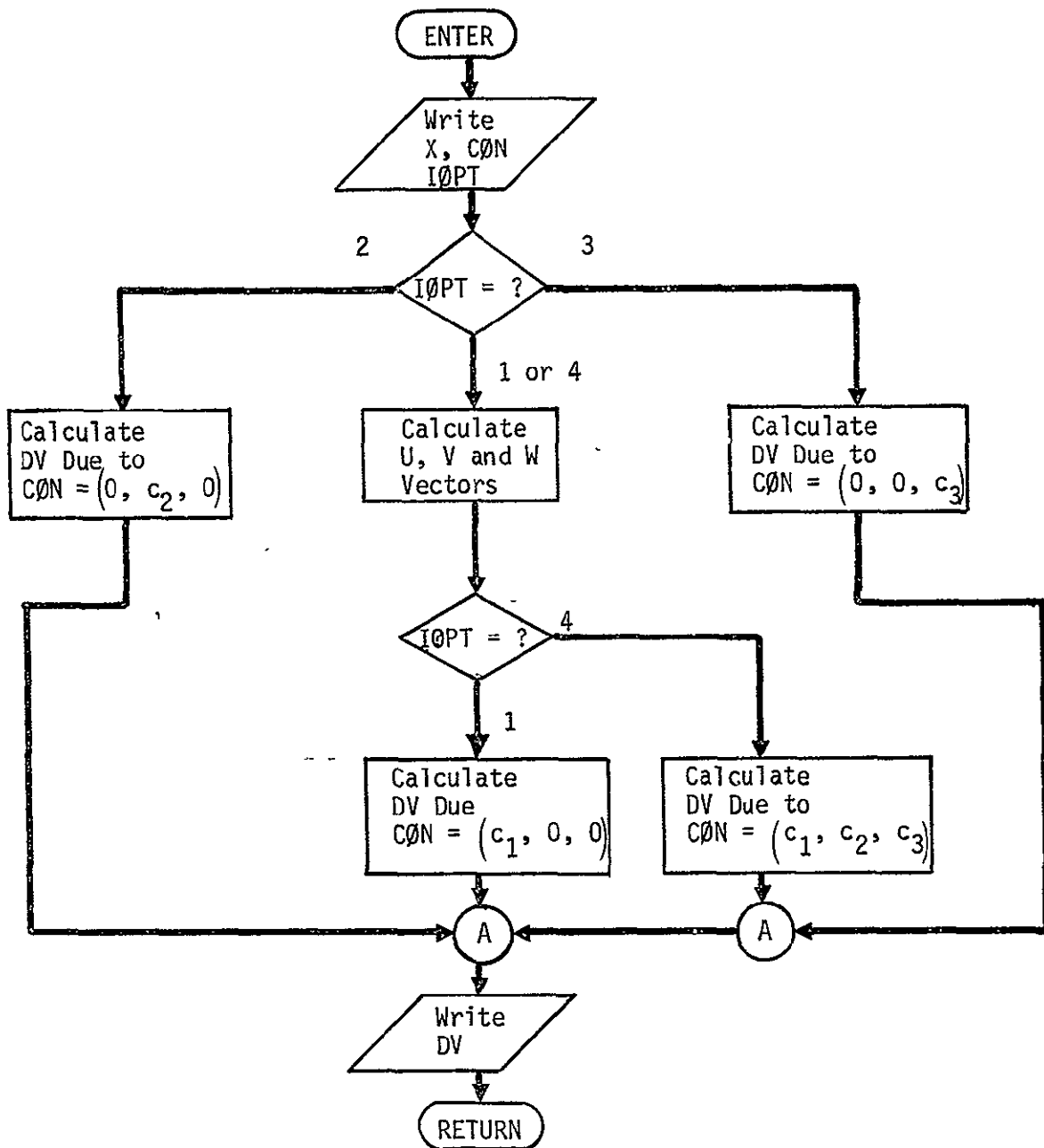
Figure 1 Pictorial Definition of Launch-Planetocentric Targeting Controls

Next consider the calculation of the increment $\Delta \underline{v}$ produced by perturbing all three controls simultaneously as required in the Newton-Raphson control correction. KTRØL performs this computation when $IØPT = 4$. Reasoning again from Figure 1

$$\Delta \underline{v} = \left[(\|\underline{v}\| + c_1) \cos c_2 \cos c_3 - \|\underline{v}\| \right] \underline{V} + (\|\underline{v}\| + c_1) \sin c_2 \cos c_3 \underline{U} + (\|\underline{v}\| + c_1) \sin c_3 \underline{W}. \quad (7)$$

Note that equation (7) degenerates to equations (4), (5), and (6) when the appropriate controls are set to zero.

KTRØL Flow Chart



SUBROUTINE LAUNCH

PURPOSE: TO COMPUTE THE INJECTION TIME, POSITION, AND VELOCITY FROM THE DEPARTURE ASYMPTOTE AND THE LAUNCH PROFILE

ARGUMENT: DI JULIAN DATE AT INJECTION (OUTPUT)
 R7 POSITION VECTOR AT INJECTION (OUTPUT)
 VZ VELOCITY VECTOR AT INJECTION (OUTPUT)

SUBROUTINES SUPPORTED: HELIO

SUBROUTINES REQUIRED: FPHEM ORR PECEQ

LOCALS SYMBOLS: ANGLE INTERMEDIATE ANGLE USED TO DEFINE TL
 AZI PLANETOCENTRIC AZUMUTH AT INJECTION (DEG)
 BHAT UNIT VECTOR NORMAL TO SHAT AND WHAT USED TO DEFINE THE P-Q ELEMENTS OF THE DEPARTURE HYPERBOLA
 BMAG MAGNITUDE OF THE NON-UNITIZED BHAT VECTOR
 COSFL COSINE OF FL
 COSFS COSINE OF FS
 COSGAM COSINE OF GAMMAI
 COSPHI COSINE OF FI
 COSSIG CONSINE OF SIGMAL
 COSHL CONSINE OF HL
 C3 VIS VIVA ENERGY ON THE DEPARTURE HYPERBOLA
 DD INTERMEDIATE VARIABLE USED TO CALCULATE GREENWICH HOUR ANGLE
 DLA PLANETOCENTRIC EQUATORIAL DECLINATION OF THE DEPARTURE ASYMPTOTE
 EQEC TRANSFORMATION MATRIX FROM ECLIPTIC TO LAUNCH PLANET EQUATORIAL
 FL TRUE ANOMALY OF LAUNCH SITE POSITION VECTOR

FS	TRUE ANOMALY OF DEPARTURE ASYMPTOTE
GAMMAI	FLIGHT PATH ANGLE AT INJECTION
GH	GREENWICH HOUR ANGLE
GMLP	GRAVITATIONAL CONSTANT OF THE LAUNCH PLANET IN KM^3/SEC^2
HE	ECCENTRICITY OF THE DEPARTURE HYPERBOLA
ID	INTERMEDIATE VARIABLE USED TO COMPUTE GREENWICH HOUR ANGLE
IHR	HOUR OF INJECTION
IMN	MINUTE OF INJECTION
I	INDEX
J	INDEX
LHR	HOUR OF LAUNCH
LMN	MINUTE OF LAUNCH
PHAT	UNIT VECTOR POINTING TOWARD PERIAPSIS OF THE HYPERBOLA
PHII	LATITUDE OF INJECTION
PSIB	THE ANGLE FROM LAUNCH TO INJECTION
QHAT	UNIT VECTOR NORMAL TO PHAT POINTING IN THE DIRECTION OF MOTION
RAI	RIGHT ASCENSION AT INJECTION
RAL	RIGHT ASCENSION OF DEPARTURE ASYMPTOTE
REFJD	JULIAN DATE FOR 1950
RIMAG	MAGNITUDE OF THE SPACECRAFT POSITION AT INJECTION
RI	SPACECRAFT POSITION AT INJECTION
RLHAT	LAUNCH SITE POSITION UNIT VECTOR
SECI	SECOND OF INJECTION
SECL	SECOND OF LAUNCH

SHAT	UNIT SPACECRAFT VELOCITY VECTOR IN EQUATORIAL SYSTEM AT INJECTION
SINFL	SINE OF FL
SINFS	SINE OF FS
SINGAM	SINE OF GAMMAI
SINPHI	SINE OF FI
SINSIG	SINE OF SIGMAL
SINWL	SINE OF WL
SLR	SIMI-LATUS RECTUM OF THE DEPARTURE HYPERBOLA
TB	TIME BETWEEN LAUNCH AND INJECTION IN SECONDS
TC	LENGTH OF PARKING ORBIT COAST IN SECONDS
TEST	INTERMEDIATE VARIABLE TO TEST FOR VIOLATION OF AZIMUTH CONSTRAINT
TFRAC	INTERMEDIATE VARIABLE USED TO CALCULATE GREENWICH HOUR ANGLE
THETAI	LONGITUDE AT INJECTION
TH	INTERMEDIATE VARIABLE USED TO CALCULATE CLOCK TIMES OF LAUNCH AND INJECTION
TI	INJECTION TIME IN DAYS REFERENCED TO MIDNIGHT OF THE LAUNCH DAY
TL	LAUNCH TIME IN DAYS REFERENCED TO MIDNIGHT OF THE LAUNCH DAY
TMN	INTERMEDIATE VARIABLE USED TO CALCULATE CLOCK TIMES OF LAUNCH AND INJECTION
TSTAR	INTERMEDIATE VARIABLE USED TO COMPUTE GREENWICH HOUR ANGLE
TWOFOR	CONSTANT VALUE, EQUAL TO 24.
VHL	MAGNITUDE OF VZ, THE INPUT VECTOR OF THE DEPARTURE ASYMPTOTE
VIMAG	MAGNITUDE OF SPACECRAFT VELOCITY AT INJECTION

WHAT UNIT VECTOR NORMAL TO THE LAUNCH PLANE IN
EQUATORIAL SYSTEM

WL RIGHT ASCENSION OF THE LAUNCH SITE

WMAG MAGNITUDE OF THE NON-UNITIZED WHAT VECTOR

XTIM INTERMEDIATE VARIABLE USED TO COMPUTE
CLOCK TIMES OF LAUNCH AND INJECTION

COMMON COMPUTED/USED: SIGNAL

COMMON COMPUTED: NO

COMMON USED:

ALNGTH	DPA	FI	FOUR	KOAST
NINETY	NLP	ONE	PHILS	PMASS
PSI1	PSI2	RAD	RAP	RPRAT
RP	THEDOT	THELS	TIM1	TIM2
TM	TWO	VHPM	XP	ZERO

LAUNCH Analysis

LAUNCH computes the injection time, position and velocity from the departure velocity \vec{v}_{HE} (computed in HELIO) and the launch profile parameters input by the user.

The rotation matrix Φ_{ECEQ} defining the transformation from ecliptic to equatorial coordinates is first computed (PECEQ). The departure velocity \vec{v}_{HE} is then normalized and converted into ecliptic coordinates to yield the departure asymptote \hat{S} .

$$\hat{S} = \Phi_{ECEQ} \frac{\vec{v}_{HE}}{v_{HE}} \quad (1)$$

Auxiliary information associated with \hat{S} is then computed. The energy C_3 , the declination ϕ_S and the right ascension θ_S of the departure asymptote, and the eccentricity of the departure hyperbola are given by

$$\begin{aligned} C_3 &= v_{HE}^2 \\ \sin \phi_S &= S_z \\ \tan \theta_S &= \frac{S_y}{S_x} \\ e &= 1 + \frac{r_p C_3}{\mu} \end{aligned} \quad (2)$$

where r_p is the desired parking orbit radius and μ is the gravitational constant of the launch planet.

The unit normal \hat{W} to the launch plane in equatorial coordinates is then computed. \hat{W} is defined by

$$\begin{aligned} W_z &= \cos \phi_L \sin \Sigma_L \\ W_y &= \frac{-W_z S_y S_z + k S_x \left[1 - (S_z^2 + W_z^2) \right]^{\frac{1}{2}}}{S_x^2 + S_y^2} \\ W_x &= \frac{-(W_y S_y + W_z S_z)}{S_x} \end{aligned} \quad (3)$$

where Φ_L is the launch site latitude, Σ_L is the launch azimuth, and $k = +1$ or -1 for the long or short coast time models respectively. The second equation defines an implicit constraint on Σ_L

$$\sin^2 \Sigma_L \leq \frac{\cos^2 \Phi_S}{\cos^2 \Phi_L} \quad (4)$$

The right ascension at launch Θ_L may now be defined by

$$\begin{aligned} \cos \Theta_L &= \frac{W_x \sin \Phi_L \sin \Sigma_L + W_y \cos \Sigma_L}{W_z^2 - 1} \\ \sin \Theta_L &= \frac{W_y \sin \Phi_L \sin \Sigma_L - W_x \cos \Sigma_L}{W_z^2 - 1} \end{aligned} \quad (5)$$

and the unit vector toward the launch position is then

$$\mathbf{R}_L = (\cos \Phi_L \cos \Theta_L, \cos \Phi_L \sin \Theta_L, \sin \Phi_L) \quad (6)$$

The complementary unit vectors $\hat{\mathbf{P}}, \hat{\mathbf{Q}}$ defining the orientation of the hyperbola within the launch plane are now introduced. Let

$$\hat{\mathbf{B}} = \hat{\mathbf{S}} \times \hat{\mathbf{W}} \quad (7)$$

The true anomaly of the departure asymptote is $\cos f_s = -\frac{1}{e}$. Then $\hat{\mathbf{P}}$ and $\hat{\mathbf{Q}}$ are given as

$$\begin{aligned} \hat{\mathbf{P}} &= \hat{\mathbf{S}} \cos f_s + \hat{\mathbf{B}} \sin f_s \\ \hat{\mathbf{Q}} &= \hat{\mathbf{S}} \sin f_s + \hat{\mathbf{B}} \cos f_s \end{aligned} \quad (8)$$

The true anomaly of the launch site f_L may now be given

$$\begin{aligned} \cos f_L &= \hat{\mathbf{R}}_L \cdot \hat{\mathbf{P}} \\ \sin f_L &= \hat{\mathbf{R}}_L \cdot \hat{\mathbf{Q}} \end{aligned} \quad (9)$$

The angle Ψ_B between launch and injection is

$$\Psi_B = 2\pi - f_L + f_I \quad (10)$$

where f_I is the desired true anomaly at injection read in as input.

The coast time t_c may now be computed from

$$t_c = \left[\Psi_B - (\Psi_1 + \Psi_2) \right] k_{\Phi} \quad (11)$$

where Ψ_1 and Ψ_2 are the angles of the first and second burns and k_{Φ} is the inverse of the parking orbit coast rate, all of which are read in as input.

The time between launch and injection is therefore

$$t_B = t_1 + t_2 + t_c \quad (12)$$

where t_1 and t_2 are the input time durations of the first and second burns.

The unit vector to injection is

$$\hat{R}_I = \hat{P} \cos f_I + \hat{Q} \sin f_I \quad (13)$$

The semi-latus rectum p is

$$p = \frac{\mu(e'^2 - 1)}{C_3} \quad (14)$$

The radius magnitude to injection is

$$R_I = \frac{p}{1 + e \cos f_I} \quad (15)$$

The injection speed is

$$V_I = \sqrt{C_3 + \frac{2\mu}{R_I}} \quad (16)$$

The path angle at injection is

$$\cos \Gamma_I = \frac{\sqrt{\mu p}}{R_I V_I} \quad (17)$$

The injection latitude is

$$\sin \phi_I = \hat{R}_{I_z} \quad (18)$$

The injection right ascension is

$$\tan \Theta_I = \frac{R_{I_y}}{R_{I_x}} \quad (19)$$

The injection longitude is

$$\Theta_I = \Theta_L + \Theta_I - \Theta_L - \omega t_B \quad (20)$$

where Θ_L is the longitude of the launch site and ω is the rotation rate of the launch planet, both being read in as input.

The injection azimuth is

$$\cos \Sigma_I = \frac{S_z - \cos(f_s - f_I) \sin \phi_I}{\sin(f_s - f_I) \cos \phi_I} \quad (21)$$

The launch time on the day of launch is

$$t_L = \frac{(\Theta_L - \Theta_I - \text{GHA}) \bmod 2\pi}{\omega} \quad (22)$$

where GHA is the Greenwich hour angle at 0^h UT of the launch date

$$\text{GHA} = 100.07554260 + 0.9856473460 T_d + 2.9015 \times 10^{-3} T_d^3 \quad (23)$$

where T_d = days past 0^h January 1, 1950.

The injection radius vector is now computed from

$$\begin{aligned} \vec{R}_I &= R_I \hat{R}_I \\ \vec{V}_I &= \frac{V_I}{R_I} \left[(\hat{W} \times \vec{R}_I) \cos \Gamma_I + \vec{R}_I \sin \Gamma_I \right] \end{aligned} \quad (24)$$

The injection time is

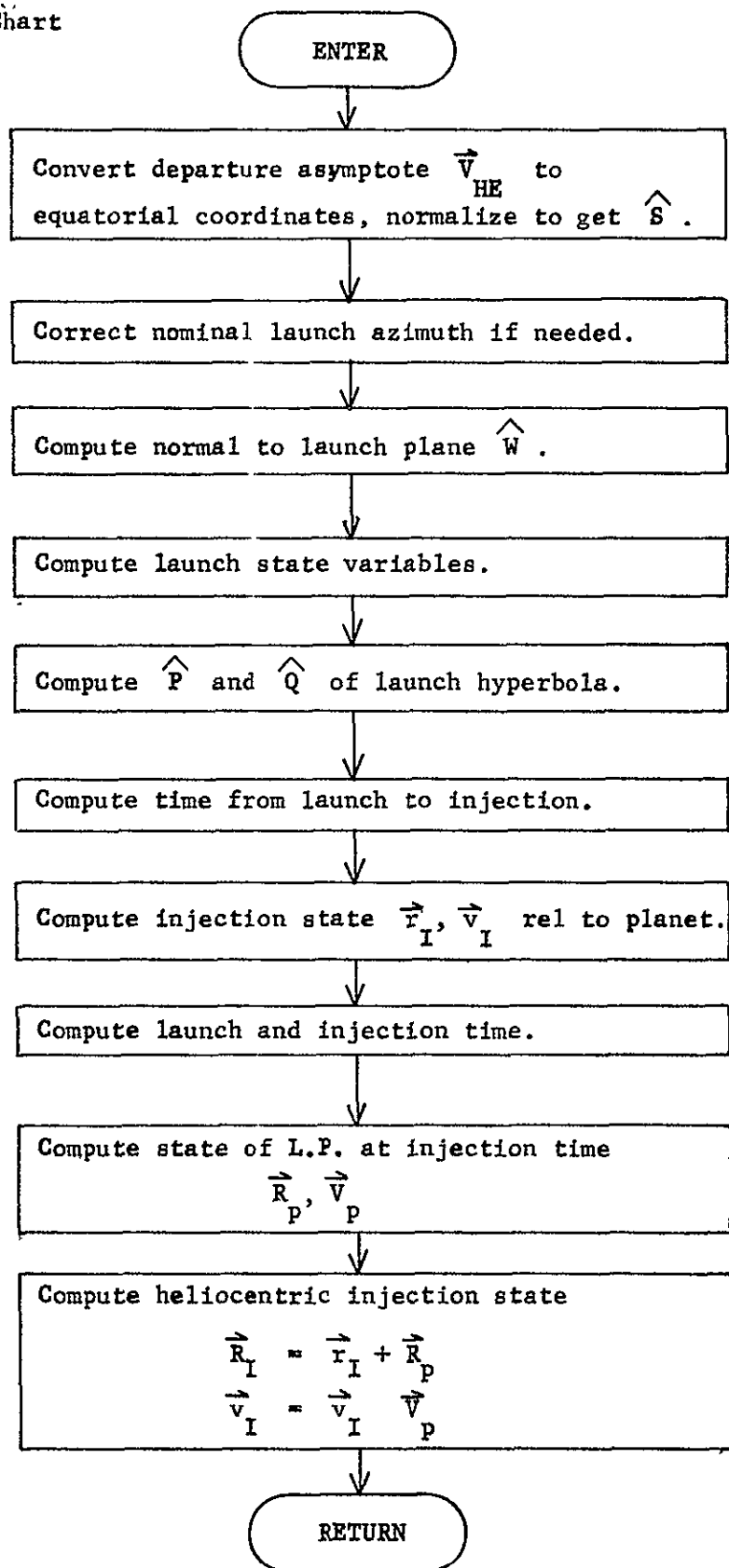
$$T_I = T_o + t_L + t_B \quad (25)$$

where T_0 is the Julian date of the launch calendar date.

The injection position and velocity are now rotated into the ecliptic plane. The position and velocity of the launch planet at the time T_I are computed and added to the injection state to get the heliocentric injection state.

Reference: Space Research Conic Program, Phase III, May 1, 1969, Jet Propulsion Laboratory, Pasadena, California.

LAUNCH Flow Chart



SUBROUTINE LUNA

PURPOSE: TO CONTROL THE GENERATION OF THE ZERO ITERATE FOR LUNAR
TARGETING

CALLING SEQUENCE: CALL LUNA

SUBROUTINES SUPPORTED: ZERIT

SUBROUTINES REQUIRED: LUNTAR MULTAR

LOCAL SYMBOLS: I INDEX

OSPH ORIGINAL SPHERE OF INFLUENCE OF TARGET
PLANET IN A.U.

COMMON COMPUTED/USED: OTAR SPHERE

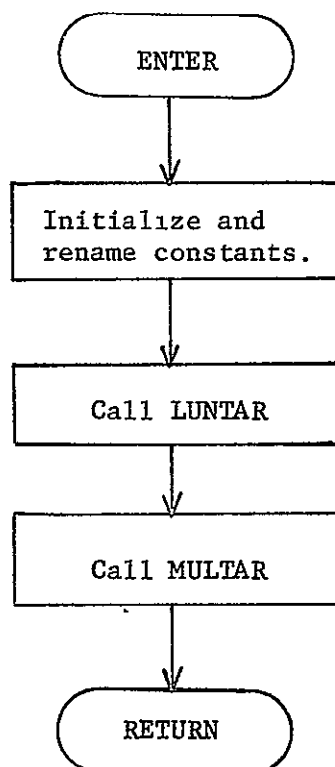
COMMON COMPUTED: BCON CAI IBARY ICOORD PCON
RCA RPE SMA TCA TSPH
TTOL

COMMON USED: ALNGTH DT FOUR KUR NTP
ONE RP SPHFAC TEN ZDAT

LUNA Analysis

LUNA is the controlling subroutine for lunar zero iterate targeting. It first serves an interface role in which it initializes constants and renames variables for the other lunar targeting routines. It then calls LUNTAR for the targeting of the lunar patched conic. When that is completed it calls MULTAR for the targeting of the multi conic trajectory. It then returns control to PRELIM.

LUNA Flow Chart



SUBROUTINE LUNCON

PURPOSE: TO COMPUTE THE ACTUAL VALUES OF THE TARGET PARAMETERS (A, BDT, BDR) FOR A LUNAR PATCHED CONIC TRAJECTORY DETERMINED BY CONTROL VALUES OF ALPHA, DELTA, AND THETA.

CALLING SEQUENCE: CALL LUNCON(ALPHAI,DELTAI,THETAI,AM,BDT,BDR,SIGNAL,ITR)

ARGUMENTS: ALPHAI I ANGLE DEFINING PERIGEE OF TRANSFER CONIC (RAD)
 DELTAI I DELINATION OF LSI POINT (RAD)
 THETAI I RIGHT ASCENSION OF LSI POINT (RAD)
 AM 0 SEMIMAJOR AXIS OF LUNAR CONIC
 BDT 0 IMPACT PARAMETER OF LUNAR CONIC
 BDR 0 IMPACT PARAMETER OF LUNAR CONIC
 SIGNAL I/O NOMINAL LAUNCH AZIMUTH OR THAT REQUIRED
 ITR 0 OUTPUT ITERATION COUNTER

SUBROUTINES SUPPORTED: LUNTAR

SUBROUTINES REQUIRED: CAREL IMPACT

LOCAL SYMBOLS: ALPHA ALPHAI IN DEGREES
 AOUT TEMPORARY LOCATION FOR AM
 CC ANGULAR MOMENTUM OF THE EARTH CENTERED TRANSFER CONIC
 CDEL COSINE OF DELTAI
 CECC ECCENTRICITY OF THE EARTH CENTERED TRANSFER CONIC
 COSDEC COSINE OF DECLIN
 COSPL COSINE OF PHIL
 COSPS INTERMEDIATE VARIABLE TO TEST FOR VIOLATION OF SIGNAL CONSTRAINT
 COSSIG COSINE OF SIGNAL
 CP SEMI-LATUS RECTUM OF EARTH CENTERED TRANSFER CONIC

CSMA	SEMI-MAJOR AXIS OF EARTH CENTERED TRANSFER CONIC
CT	COSINE OF THETA I
DELTA	DELTA I IN DEGREES
EM	ECCENTRICITY OF LUNAR CONIC
GAMMI	INTERMEDIATE ANGLE USED TO COMPUTE EARTH CENTERED TRANSFER CONIC
I	INDEX
PHIL	LATITUDE OF LAUNCH SITE
POS	SPACECRAFT POSITION AND VELOCITY AT LSI POINT IN MOON-CENTERED EARTH EQUATORIAL COORDINATES
PPH	DUMMY VARIABLE FOR CALL CAREL
QQH	DUMMY VARIABLE FOR CALL TO CAREL
RAD	RADIANS TO DEGREES CONVERSION FACTOR
RMAG	MAGNITUDE OF THE RI VECTOR
ROUT	VELOCITY AT LSI IN GEOCENTRIC EQUATORIAL SYSTEM
RPM	RADIUS OF PERIAPSIS OF LUNAR CONIC
SDEL	SINE OF DELTA I
SHAT	UNIT VECTOR POINTING FROM THE EARTH TO THE POINT DEFINED BY DELTA I, THETA I
SIGN	SIGNAL IN DEGREES
SINDEC	SINE OF DECLIN
SINPS	INTERMEDIATE VARIABLE USED TO TEST FOR VIOLATION OF SIGNAL CONSTRAINT
SINSIG	SINE OF SIGNAL
SX	SINE OF THETA I
TAM	TRUE ANOMALY OF THE LUNAR CONIC CORRESPONDING TO THE RSI VECTOR

TFLP TIME OF FLIGHT FROM PERIAPSIS CORRESPOND-
 ING TO THE RSI VECTOR

 THETA THETA I IN DEGREES

 VMAG SPACECRAFT VELOCITY MAGNITUDE USED TO
 CALCULATE DECLIN

 WHAT UNIT VECTOR NORMAL TO THE EARTH-PHASE

 WMAG ANGULAR MOMENTUM CONSTANT

 WM ARGUMENT OF PERIAPSIS OF THE LUNAR CONIC

 WWM DUMMY VARIABLE FOR CALL TO CAREL

 XHAT SAME AS SHAT VECTOR

 XIM INCLINATION OF THE CONIC

 XNM LONGITUDE OF THE ASCENDING NODE OF THE
 LUNAR CONIC

 YHAT CROSS PRODUCT OF THE WHAT AND XHAT VECTORS

COMMON COMPUTED/USED:

RI RSI

COMMON COMPUTED:

DECLIN

COMMON USED:

EMU	EQLQ	KOAST	NINETY	ONE
PHILS	RMQ	RPE	SIGMA	TMU
TSPH	TWO	ZERO		

LUNCON Analysis

The point of intersection of the Earth-centered conic with the lunar sphere of influence (LSI) is determined by the angles θ and δ . Relative to the moon in Earth-equatorial coordinates that point is

$$\vec{r}_{SI} = \begin{bmatrix} R_{SI} \cos \delta \cos \theta \\ R_{SI} \cos \delta \sin \theta \\ R_{SI} \sin \delta \end{bmatrix} \quad (1)$$

where R_{SI} is the radius of the LSI. Relative to the earth that point is

$$\vec{R}_q = \vec{R}_M + \vec{r}_{SI} \quad (2)$$

where \vec{R}_M is the radius vector to the center of the moon at the time of LSI intersection t_{SI} in earth equatorial coordinates.

There are at most two planes which contain \vec{R}_q and satisfy the launch latitude ϕ and azimuth Σ constraints. Let \hat{W} denote the unit normal to either of these planes. Now let \hat{R}_L , θ_L , ϕ_L denote the unit vector, longitude, and latitude of the launch site. Construct a local horizon coordinate system at the launch site as indicated in Figure 1.

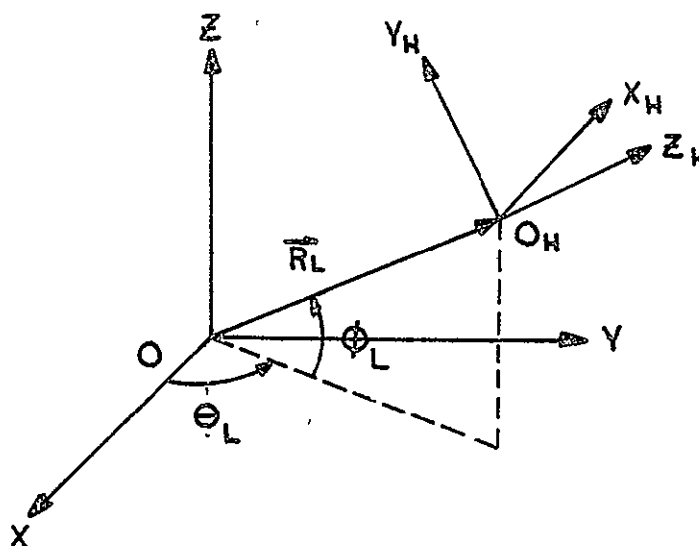


Figure 1. Local Horizon Coordinate System

Here $\hat{Z}_h = \frac{\hat{R}_L}{R_L}$, \hat{Y}_L is normal to \hat{Z}_h in the \hat{Z} -O- \hat{R}_L plane, and $\hat{X}_h = \hat{Y}_h \times \hat{Z}_h$.

In the local horizon system, the position and velocity are very simply represented

$$\begin{aligned}\vec{R}_h &= R [0, 0, 1]^T \\ \vec{V}_h &= V [\cos \delta \sin \Sigma, \cos \delta \cos \Sigma, \sin \delta]^T\end{aligned}\quad (3)$$

where Σ is the launch azimuth and δ is the declination wrt the local horizontal. Thus

$$\hat{W}_h = \frac{\vec{R}_h \times \vec{V}_h}{|\vec{R}_h \times \vec{V}_h|} = \begin{bmatrix} -\cos \Sigma \\ \sin \Sigma \\ 0 \end{bmatrix} \quad (4)$$

The transformation matrix converting a vector in the local horizon system to the equatorial system is

$$T = \begin{bmatrix} -\sin \theta_L & -\sin \phi_L \cos \theta_L & \cos \phi_L \cos \theta_L \\ \cos \theta_L & -\sin \phi_L \sin \theta_L & \cos \phi_L \sin \theta_L \\ 0 & \cos \phi_L & \sin \phi_L \end{bmatrix} \quad (5)$$

Therefore since $\vec{W}_q = T \vec{W}_h$, the z-component of \hat{W} in the equatorial coordinate system is

$$\hat{W}_z = \cos \phi_L \sin \Sigma \quad (6)$$

Since \hat{W} is a unit normal it must satisfy both $\hat{W} \cdot \hat{W} = 1$ and $\hat{W} \cdot \hat{S} = 0$ where $\hat{S} = \frac{\vec{R}_q}{R_q}$. Solving for the two remaining components of \hat{W} ,

$$\hat{W}_y = \frac{-\hat{W}_z \hat{S}_y \hat{S}_z \pm \hat{S}_x \sqrt{1 - (\hat{S}_z^2 + \hat{W}_z^2)}}{\hat{S}_x^2 + \hat{S}_y^2} \quad (7)$$

$$\hat{W}_x = \frac{-(\hat{W}_y \hat{S}_y + \hat{W}_z \hat{S}_z)}{\hat{S}_x} \quad (8)$$

To eliminate the ambiguity of sign in (7) the short-coast plane corresponding to the negative sign is used. Note that (7) also imposes a constraint on the launch azimuth

$$\sin^2 \Sigma \leq \frac{1 - \hat{S}_z^2}{\cos^2 \varphi_L} \quad (9)$$

Now choose $\hat{U} = \hat{W} \times \hat{S}$ to complete a right hand system $(\hat{S}, \hat{U}, \hat{W})$. Then the position at LSI relative to the earth is $(R_I, 0, 0)$. Now let α determine the perigee point in the orbital plane ($\hat{W} = 0$) measured counterclockwise from the $-\hat{S}$ axis. Then the perigee point is $(-r_p \cos \alpha, -r_p \sin \alpha, 0)$ where r_p is the parking orbit radius (input). Therefore the true anomaly of the earth centered conic at the LSI is given by

$$f_{SI} = 180 - \alpha \quad (10)$$

The two equations $R_I = \frac{a(1 - e^2)}{1 + e \cos f_{SI}}$ and $r_p = a(1 - e)$ may be solved simultaneously for the semi-major axis a and eccentricity e of the unique earth centered conic

$$e_g = \frac{R_I - r_p}{r_p - R_I \cos f_{SI}} \quad (11)$$

$$a_g = \frac{r_p}{1 - e_g} \quad (12)$$

Thus the velocity of the earth centered conic at the LSI is in the $(\hat{S}, \hat{U}, \hat{W})$ system

$$\vec{V}_o = \begin{bmatrix} \sqrt{a(1-e^2)} & e \sin f_{SI} \\ \mu a(1-e^2) / R_I \\ 0 \end{bmatrix} \quad (13)$$

Transforming to the earth equatorial coordinate system

$$\vec{V}_q = \begin{bmatrix} S_x & U_x & W_x \\ S_y & U_y & W_y \\ S_z & U_z & W_z \end{bmatrix} \vec{V}_o \quad (14)$$

Now if $(\vec{R}_{MQ}, \vec{V}_{MQ})$ are the position and velocity of the moon at t_{SI} Earth-centered coordinates and (\vec{R}_Q, \vec{V}_Q) are the position and velocity of the spacecraft at t_{SI} then the state of the spacecraft with respect to the moon at t_{SI} is in earth equatorial coordinates

$$\begin{aligned}\vec{r}_{SI} &= \vec{R}_Q - \vec{R}_{MQ} \\ \vec{v}_{SI} &= \vec{V}_Q - \vec{V}_{MQ}\end{aligned}\tag{15}$$

Using the transformation matrix ϕ_{EQLQ} defining transformations from earth equatorial to lunar equatorial the state in the LQ system is

$$\begin{aligned}\vec{r}_{sq} &= \phi_{EQLQ} \vec{r}_{SI} \\ \vec{v}_{sq} &= \phi_{EQLQ} \vec{v}_{SI}\end{aligned}\tag{16}$$

The impact plane parameters $B \cdot T$ and $B \cdot R$, and the inclination i_ℓ , may now be computed by calling subroutines ACTB and CAREL.

SUBROUTINE LUNTAR

PURPOSE: TO GENERATE A PATCHED CONIC TRAJECTORY FOR LUNAR MISSIONS CONSISTENT WITH TARGET PARAMETERS AT THE MOON OF (ACA, RCA, ICA, TCA) AND LAUNCH PARAMETERS (PHIL, THETA, SIGMA).

CALLING SEQUENCE: CALL LUNTAR

SUBROUTINES SUPPORTED: LUNA

SUBROUTINES REQUIRED: LUNCON EPHM IMPACT MATIN ORB
PECEQ

LOCAL SYMBOLS: AA SEMI-MAJOR AXIS OF THE LUNAR CONIC FOR THE
NOMINAL TRAJECTORY

ALNGTH SAME AS AU

ALPHA1 REFINED ANGLE (RADIAN) DEFINING POSITION
OF PERIGEE ON THE TRANSFER CONIC
(NOMINALLY SET TO FIVE DEGREES)

ALPI PERTURBED VALUE OF ALPHA1 USED TO SOLVE
FOR RCA, ICA, ACA

AUDAY CONVERTS KM/SEC TO AU/DAY

AUS SAME AS AU

AU CONVERTS KILOMETERS (KM) TO ASTRONOMICAL
UNITS (AU)

BDR B DOT R FOR THE NOMINAL TRAJECTORY

BDT B DOT T FOR THE NOMINAL TRAJECTORY

BINC OBTAINABLE INCLINATION USED TO CALCULATE
DESIRED B DOT T, B DOT R

DELI PERTURBED VALUE OF DELTA1 USED TO SOLVE
FOR RCA, ICA, ACA

DELTA1 REFINED ANGLE (RADIAN) DEFINING DECLIN-
ATION OF THE LSI POINT (NOMINALLY SET TO
DELTA0)

DELTA0 DECLINATION OF THE MOONS POSITION AT TIME
TSI

DELT TIME FROM TSI TO TCA IN SECONDS

DEL REFINING VALUES FOR ALPHA1, DELTA1, THETA1

DENOM	INTERMEDIATE VARIABLE USED TO LIMIT THE DEL VALUES FOR EACH ITERATION
ECC	DESIRED ECCENTRICITY OF THE LUNAR CONIC
ECEQ	TRANSFORMATION MATRIX FROM ECLIPTIC TO EARTH EQUATORIAL
ECLQ	TRANSFORMATION MATRIX FROM ECLIPTIC TO LUNAR EQUATORIAL
ERR	VECTOR OF DIFFERENCES BETWEEN DESIRED AND NOMINAL VALUES OF $B \cdot T$, $B \cdot R$, ACA
ITAR	LOGIC CONTROLLING INDICATOR =1 IMPROVE ACA ONLY =2 IMPROVE RCA, ICA, ACA
ITER	ITERATION COUNTER FOR NOMINAL TRAJECTORIES
IT	ITERATION COUNTER FOR PERTURBED TRAJECT- ORIES
I	INDEX
J	INDEX
K	INDEX
ONEMAT	UNIT DUMMY MATRIX FOR CALL TO IMPACT
PAI	DUMMY VARIABLE FOR CALL TO LUNCON WHEN ITAR=1
PARP	DUMMY VARIABLE FOR CALL TO LUNCON WHEN ITAR=1
PARTA	INTERMEDIATE VARIABLE USED TO REFINE ACA WHEN ITAR=1
PARTW	INTERMEDIATE VARIABLE USED TO COMPUTE DELT
PARTX	INTERMEDIATE VARIABLE USED TO COMPUTE DELT
PARTY	INTERMEDIATE VARIABLE USED TO COMPUTE DELT
PARTZ	INTERMEDIATE VARIABLE USED TO COMPUTE DELT
PHI	MATRIX RELATING PERTURBATIONS IN ALPHA _I , DELTA _I , AND THETA _I TO CHANGES IN $B \cdot T$, $B \cdot R$, AND ACA

PSI TARGETING MATRIX RELATING PERTURBATIONS
 IN δ DOT T, δ DOT R, AND ACA TO CHANGES
 IN ALPHA_I, DELTA_I, AND THETA_I

 PTAR PERTURBED VALUES OF AA, BDT, BDR, USED TO
 CALCULATE PHI

 RAD CONVERTS DEGREES TO RADIAN

 RMAG MAGNITUDE OF THE RMQ VECTOR

 SIGNAL LAUNCH AZIMUTH SET IN LUNCON (NOMINALLY
 90 DEGREES)

 TAR NOMINAL VALUES OF AA, BDT AND BDR USED TO
 CALCULATE PHI

 THEI PERTURBED VALUES OF THETA_I USED TO SOLVE
 FOR RCA, ICA, AND ACA

 THETA_I REFINED ANGLE (RADIAN) DEFINING RIGHT
 ASCENSION OF THE LSI POINT (NOMINALLY SET
 TO THETA₀)

 THETA₀ RIGHT ASCENSION OF THE MOON'S POSITION AT
 TIME TSI

 TM CONSTANT VALUE OF SECONDS PER DAY

 TSICA DUMMY ARGUMENT FOR CALL TO IMPACT

COMMON COMPUTED/USED:

DTAR	EQLQ	ITAG	RMQ	RSI
TMU	TSI			

COMMON COMPUTED:

EMU	NO	RME
-----	----	-----

COMMON USED:

BCON	DECLIN	FIVE	ONE	OTAR
PCON	PHASS	RCA	SMA	TCA
TSPI	TTOL	TWO	XP	ZERO

LUNTAR Analysis

LUNTAR generates a patched conic trajectory arriving at closest approach to the Moon at a specified time t_{CA} and meeting prescribed target values at that point as well as standard launch quantities. The target parameters are

t_{CA}	Julian date of required closest approach (CA) referenced 1900
r_{CA}	Radius of CA
i_{CA}	Inclination (relative to lunar equator) at CA ¹
a_{CA}	Semi-major axis at CA

The launch parameters

ϕ_L	Launch site latitude
θ_L	Launch site longitude
Σ_L	Launch azimuth (nominally set to 90°)
r_p	Parking orbit radius

The eccentricity of the moon-centered hyperbola may be computed

$$e_{CA} = 1 - \frac{r_{CA}}{a_{CA}} \quad (1)$$

where $a_{CA} < 0$. The hyperbolic time Δt to go from R_{SI} (radius of lunar sphere of influence (LSI)) to periapsis may be computed from

$$\Delta t = f(\mu_M, a_{CA}, e_{CA}, R_{SI}) \quad (2)$$

where μ_M is the lunar gravitational constant. The time at which the probe should intersect the LSI is then

$$t_{SI} = t_{CA} - \Delta t \quad (3)$$

¹ The inclination must be specified according to the format described in IMPACT. For $0 \leq i < 90^\circ$ the inclinations $\pm i$ prescribe posigrade orbits while $180^\circ \pm i$ define retrograde orbits. The positive signs denote approaches from the north, the negative signs designate southern approaches.

The position \vec{R}_{ME} and velocity \vec{V}_{ME} of the moon at t_{SI} relative to the earth in earth ecliptic (EC) coordinates are computed by calling ORB and EPHEM. Transformation matrices ϕ_{ECEQ} and ϕ_{EQLQ} defining transformations from EC to EQ (earth equatorial) and EQ to LQ (lunar equatorial) respectively are then computed by PECEQ. The position and velocity of the moon in the EQ system are

$$\begin{aligned}\vec{R}_{MQ} &= \phi_{ECEQ} \vec{R}_{ME} \\ \vec{V}_{MQ} &= \phi_{ECEQ} \vec{V}_{ME}\end{aligned}\quad (4)$$

Call the point of intersection of the vector \vec{R}_{MQ} with the LSI the bullseye point. Then in moon-centered Earth-equatorial coordinates the vector to the bullseye point is given by

$$\vec{r}_B = -\left(\frac{\vec{R}_{MQ}}{R_{MQ}}\right) R_{SI} \quad (5)$$

From this vector one can calculate a set of angular coordinates (δ_o, θ_o) of the bullseye point. Any other point on the LSI is determined by giving general coordinates $(\delta, \theta) = (\delta_o + \Delta\delta, \theta_o + \Delta\theta)$.

Now let such a set of coordinates be given. They determine a vector \vec{R}_I from earth to the LSI (in the EQ-system). The vector \vec{R}_I along with the launch parameters $\phi_L, \theta_L, \Sigma_L$ then determines the plane of the Earth-LSI transfer (see LUNCON). Now let α be measured counter-clockwise in that plane from $-\vec{R}_I$. The parameter α specifies the location of the perigee point of the transfer conic, thus the vector to perigee is fixed as \vec{r}_p where the perigee magnitude r_p is fixed as input. The vectors \vec{r}_p and \vec{R}_I then determine a unique conic for the Earth-LSI phase (see LUNCON). Let the state at the LSI on that conic (relative to Earth-equatorial coordinates) be denoted by \vec{R}_I, \vec{V}_I . The state relative to the moon may then be computed as

$$\begin{aligned}\vec{r}_I &= \vec{R}_I - \vec{R}_{MQ} \\ \vec{v}_I &= \vec{V}_I - \vec{V}_{MQ}\end{aligned}\quad (6)$$

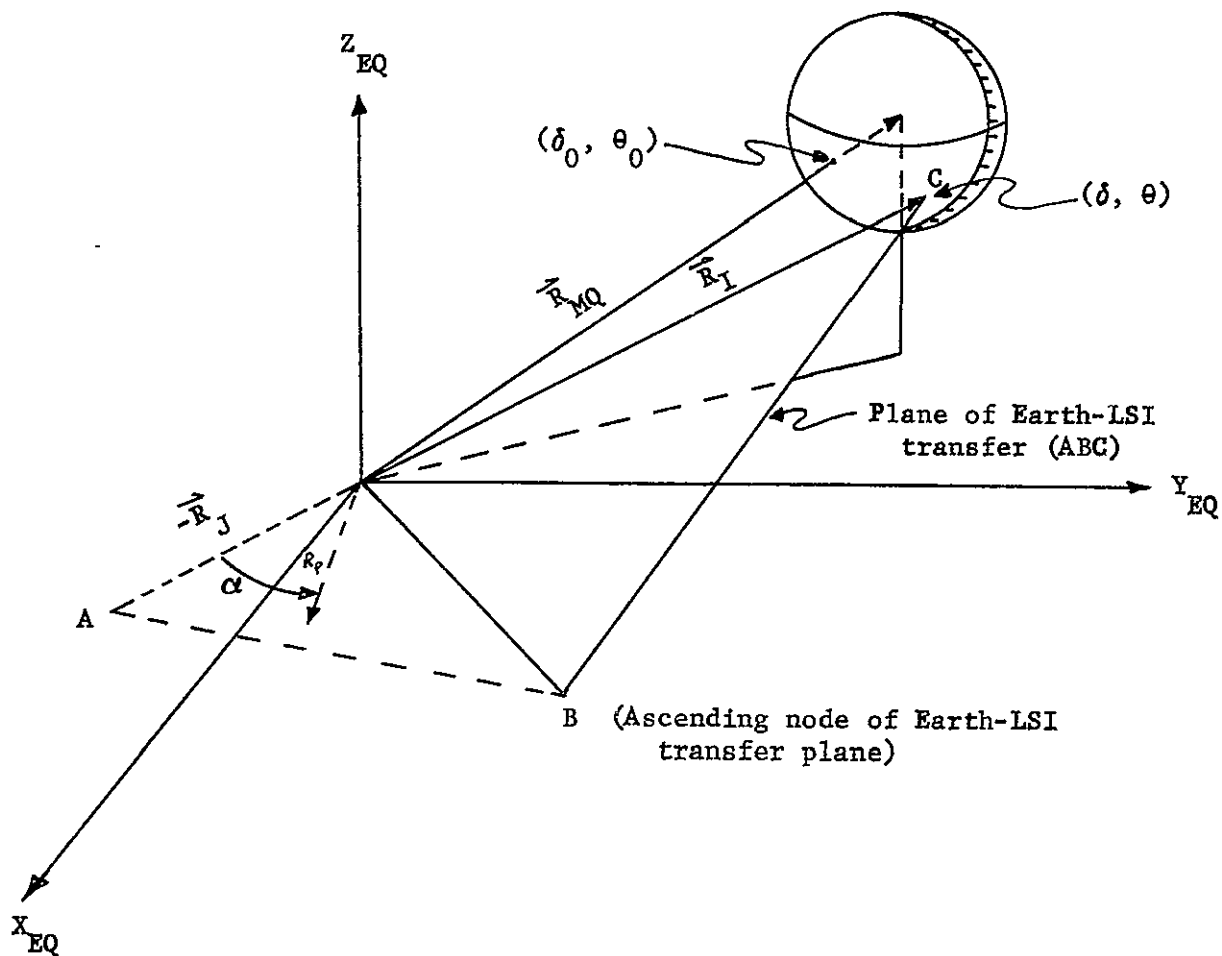


Figure 1. Lunar Patched Conic Targeting

Thus the elements relative to the moon may be computed from standard conic formula. The three angles (δ , θ , α) form a set of independent controls to be varied to meet the three constraints (r_{CA} , i_{CA} , a_{CA}). The controls are depicted in Figure 1.

LUNTAR uses the standard Newton-Raphson algorithm to refine the controls to meet the constraints. This targeting is done in two stages. In the first stage the controls δ and θ are held fixed at the bullseye point (δ_0, θ_0) while α is varied until the semi-major axis target a_{CA} is met. Then all three controls are varied to satisfy the three target constraints. The preliminary targeting of a_{CA} is essential to the success of the procedure. Once the initial targeting is completed, the semi major axis of future

iterations in the second stage will not vary much from the target value a_{CA} . For such iterates the excess hyperbolic velocity at the moon will be generally constant. This permits the substitution of the auxiliary impact plane parameters $B \cdot T$ and $B \cdot R$ for the less linear parameters of r_{CA} and i_{CA} (see IMPACT). In LUNTAR the impact plane parameters are referenced to the IQ system.

The procedure may now be described in detail. Suppose that in the first stage of targeting the current value of α is α_K . Using the controls $(\alpha_K, \delta_0, \theta_0)$ the resulting semi-major axis is found to be a_K (LUNCON). A perturbed value for the first control is then used $(\alpha_K + \Delta\alpha, \delta_0, \theta_0)$ producing a perturbed value of semi-major axis $(a_K + \Delta a)$. The $(k+1)$ st value of α is then given by the standard numerical differencing approximation

$$\alpha_{K+1} = \alpha_K + \frac{\Delta\alpha}{\Delta a} (a_{CA} - a_K) \quad (7)$$

The second stage of the targeting of the lunar patched conic uses the vector analogue of the above procedure. The current iterate $(\alpha_K, \delta_K, \theta_K)$ is input to LUNCON to obtain the current target values $(a_K, B \cdot T_K, B \cdot R_K)$.

The target values $B \cdot T$ and $B \cdot R$ are determined from subroutine IMPACT and the errors of the k th iterate are computed (e_a, e_{BT}, e_{BR}) . If all three errors are within tolerances, the procedure is terminated. Otherwise the sensitivity matrix ϕ is computed by numerical differencing as in the first stage

$$\phi = \begin{bmatrix} \frac{\Delta a_{\alpha}}{\Delta \alpha} & \frac{\Delta a_{\delta}}{\Delta \delta} & \frac{\Delta a_{\theta}}{\Delta \theta} \\ \frac{\Delta B \cdot T_{\alpha}}{\Delta \alpha} & \frac{\Delta B \cdot T_{\delta}}{\Delta \delta} & \frac{\Delta B \cdot T_{\theta}}{\Delta \theta} \\ \frac{\Delta B \cdot R_{\alpha}}{\Delta \alpha} & \frac{\Delta B \cdot R_{\delta}}{\Delta \delta} & \frac{\Delta B \cdot R_{\theta}}{\Delta \theta} \end{bmatrix} \quad (8)$$

The inverse of ϕ is the targeting matrix. The $k+1$ iterate is then defined to be

$$\begin{bmatrix} \alpha \\ \delta \\ \theta \end{bmatrix}_{K+1} = \begin{bmatrix} \alpha \\ \delta \\ \theta \end{bmatrix}_K + \phi^{-1} \begin{bmatrix} a_{CA} - a_K \\ B \cdot T - B \cdot T_K \\ B \cdot R - B \cdot R_K \end{bmatrix} \quad (9)$$

This procedure is repeated until convergence is achieved.

PROGRAM MAIN

PURPOSE: TO CONTROL THE SIMULATION OVERLAY. SCHEME

SUBROUTINES SUPPORTED: NONE

SUBROUTINES REQUIRED: DATAS SIMUL PRNTS4

LOCAL SYMBOLS: IRUNX TOTAL NUMBER OF DATA CASES

IRUN DATA CASE COUNTER

SUBROUTINE MATIN

PURPOSE: TO COMPUTE THE INVERSE OF A MATRIX.

CALLING SEQUENCE: CALL MATIN(A,R,N)

ARGUMENTS A(N,N) I MATRIX TO BE INVERTED

R(N,N) O RESULTANT INVERSE OF MATRIX A

N I DIMENSION OF A AND R

SUBROUTINES SUPPORTED: HYELS NAVM BIAIM POICOM GUISS
TARMAX GUID LUNTAR MULTAR GAIN1
GAIN2 GAUSLS TPRTRG

SUBROUTINES REQUIRED: NONE

LOCAL SYMBOLS: AL A(LL) + S (INTERMEDIATE VARIABLE)

ALBAR INTERMEDIATE VARIABLE

B INTERMEDIATE VECTOR

DETR INTERMEDIATE VECTOR

G INTERMEDIATE VECTOR

IX INTERMEDIATE VECTOR

KR DIMENSION OF A

MIXI INTERMEDIATE VARIABLE

MIXJ INTERMEDIATE VARIABLE

MIXL INTERMEDIATE VARIABLE

S INTERMEDIATE VARIABLE

X INTERMEDIATE VARIABLE

XOFF INTERMEDIATE VARIABLE

COMMON USED: EM7 EM9 ONE ZERO

SUBROUTINE MATPY

PURPOSE: T6 MULTIPLY TWO MATRICES OF REAL-VALUED ELEMENTS

ARGUMENT: A I PRE-FACTOR MATRIX
B I POST-FACTOR MATRIX
C O PRODUCT MATRIX
L I NUMBER OF ROWS OF A
M I NUMBER OF COLUMNS OF A (=NUMBER OF ROWS OF B)
N I NUMBER OF COLUMNS OF B

SUBROUTINES SUPPORTED: DIMPCP IMPCT TPPROP TPRTRG

LOCAL SYMBOLS: II INDEX USED IN ADDRESSING ELEMENTS OF C
TREATED AS COLUMN VECTOR
JJ INDEX USED IN ADDRESSING ELEMENTS OF A
TREATED AS COLUMN VECTOR
KK INDEX USED IN ADDRESSING ELEMENTS OF B
TREATED AS COLUMN VECTOR

SUBROUTINE MEAN

PURPOSE: TO PROPAGATE AND UPDATE MEANS OF ACTUAL STATE AND
PARAMETER DEVIATIONS AND ACTUAL STATE AND PARAMETER
ESTIMATION ERRORS

CALLING SEQUENCE: CALL MEAN(EXTP,EXSTP,IFLAG,IFLAG1,NR)

ARGUMENTS: EXTP I STATE DEVIATIONS OR ESTIMATION ERRORS
EXSTP I SOLVE-FOR PARAMETER DEVIATIONS OR
ESTIMATION ERRORS
IFLAG I =1 FOR UPDATE
=2 FOR PROPAGATION
IFLAG1 I =1 FOR DEVIATION MEANS
=2 FOR ESTIMATION ERROR MEANS
NR I NUMBER OF ROWS IN THE OBSERVATION MATRIX

SUBROUTINES SUPPORTED: ERRANN SETEVN PROBE GENGID PRED

LOCAL SYMBOLS: IGO INTERNALLY SET FLAG
SUM INTERMEDIATE STORAGE
ZERO VALUE 0.0.

COMMON COMPUTED/USED: DUME EU EV EW EXIP
EXSIP

COMMON USED: AK AL AM AN G
H NDIM1 NDIM2 NOIM3 NOIM4
PHI S TXU TXW TXXS

MEAN Analysis

Subroutine MEAN propagates and updates actual estimation error means over the time interval $[t_k, t_{k+1}]$ separating two successive measurements or events. The equations programmed in MEAN are independent of the filter algorithm employed to generate gain matrices. Gain matrices are assumed to have been computed during a prior call to subroutine GNAVM. The propagation equations programmed in MEAN are also used to propagate actual deviation means over the time interval separating two successive guidance events. The update equations, of course, are not used in this situation.

The actual estimation errors for position/velocity state, solve-for parameters, dynamic consider parameters, measurement consider parameters, and ignore parameters are defined, respectively, by the following:

$$\tilde{x}_{k+1} = \hat{x}_{k+1} - x_{k+1} \quad (1)$$

$$\tilde{x}_{s_{k+1}} = \hat{x}_{s_{k+1}} - x_{s_{k+1}} \quad (2)$$

$$\tilde{u}_{k+1} = \hat{u}_{k+1} - u_{k+1} = -u_o \quad (3)$$

$$\tilde{v}_{k+1} = \hat{v}_{k+1} - v_{k+1} = -v_o \quad (4)$$

$$\tilde{w}_{k+1} = \hat{w}_{k+1} - w_{k+1} = -w_o \quad (5)$$

where $(\hat{})$ indicates estimated values, and x, x_s, u_o, v_o , and w_o are the actual deviations from nominal.

Only the means of \tilde{x} and \tilde{x}_s are propagated and updated since the means of y, x , and w are constant. The propagation equations are summarized

$$E[\tilde{x}_{k+1}^-] = \Phi \cdot E[\tilde{x}_k^+] + \theta_{xx_s} \cdot E[\tilde{x}_{s_k}^+] - \theta_{xu} \bar{u}_o - \theta_{xw} \bar{w}_o \quad (6)$$

$$E[\tilde{x}_{s_{k+1}}^-] = E[\tilde{x}_{s_k}^+] \quad (7)$$

where $\Phi, \theta_{xx_s}, \theta_{xu}$, and θ_{xw} are state transition matrices over $[t_k, t_{k+1}]$.

Before the means of x and x_s can be updated at a measurement, the mean of the measurement residual ϵ_{k+1} must first be computed using

$$E[\epsilon_{k+1}] = -H \cdot E[\tilde{x}_{k+1}^-] - M \cdot E[\tilde{x}_{s_{k+1}}^-] + G\bar{u}_0 + L\bar{v}_0 + N\bar{w}_0 \quad (8)$$

where H , M , G , L , and N are observation matrix partitions.

The update equations are summarized as:

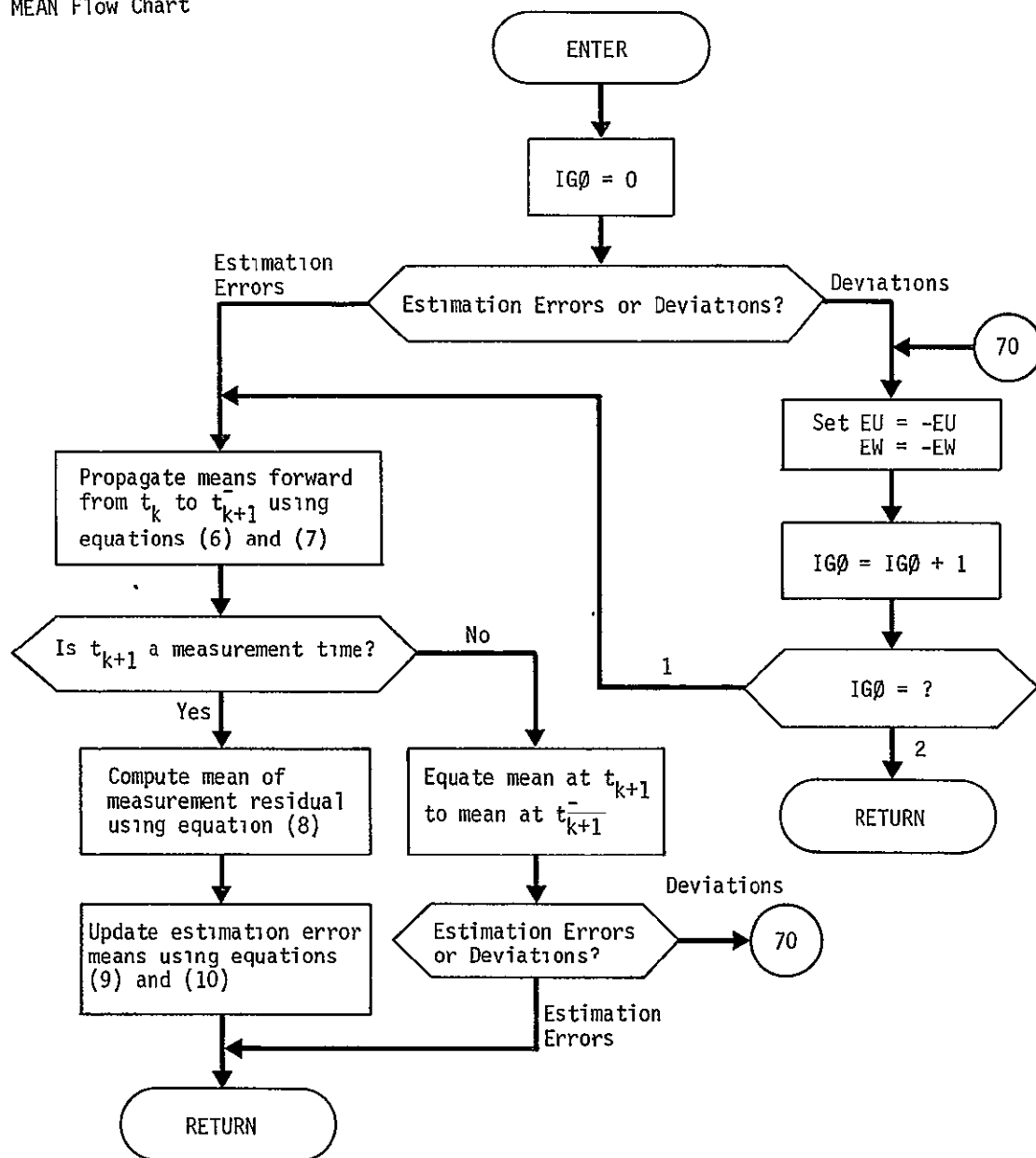
$$E[\tilde{x}_{k+1}^+] = E[\tilde{x}_{k+1}^-] + K_{k+1} \cdot E[\epsilon_{k+1}] \quad (9)$$

$$E[\tilde{x}_{s_{k+1}}^+] = E[\tilde{x}_{s_{k+1}}^-] + S_{k+1} \cdot E[\epsilon_{k+1}] \quad (10)$$

where K_{k+1} and S_{k+1} are the filter gain matrices.

- To propagate actual deviation means requires that x and x_s be replaced by \tilde{x} and \tilde{x}_s , respectively, in equations (6) and (7), and that the minus signs in equations (6) be replaced with plus signs.

MEAN Flow Chart



SUBROUTINE MENO

PURPOSE: COMPUTE ASSUMED AND ACTUAL MEASUREMENT NOISE COVARIANCE
MATRICES IN THE ERROR ANALYSIS PROGRAM

CALLING SEQUENCE: CALL MENO(MMCODE,ICODE)

ARGUMENT: ICODE I INTERNAL CODE USED TO DISTINGUISH BETWEEN
THE TWO ALTERNATIVES LISTED ABOVE

MMCODE I MEASUREMENT MODEL CODE

SUBROUTINES SUPPORTED: ERRANN

COMMON COMPUTED: R RPR

COMMON USED: IMNF MNCN IGMNF GMNCN

MENØ Analysis

The linearized observation equation employed by the navigation process is given by

$$\delta Y_k = H_k^A \delta X_k^A + \eta_k$$

where δY_k is the measurement deviation from the nominal measurement, H_k^A is the augmented observation matrix, δX_k^A is the augmented state deviation from the nominal augmented state, and η_k is the assumed measurement noise.

The function of subroutine MENØ is to compute the assumed measurement noise covariance matrix

$$R_k = E \begin{bmatrix} \eta_k & \eta_k^T \end{bmatrix}$$

if ICØDE = 0. The constant measurement noise variances associated with all available measurement types are stored in the vector MNCN. Subroutine MENØ selects the appropriate element from this vector to construct R_k .

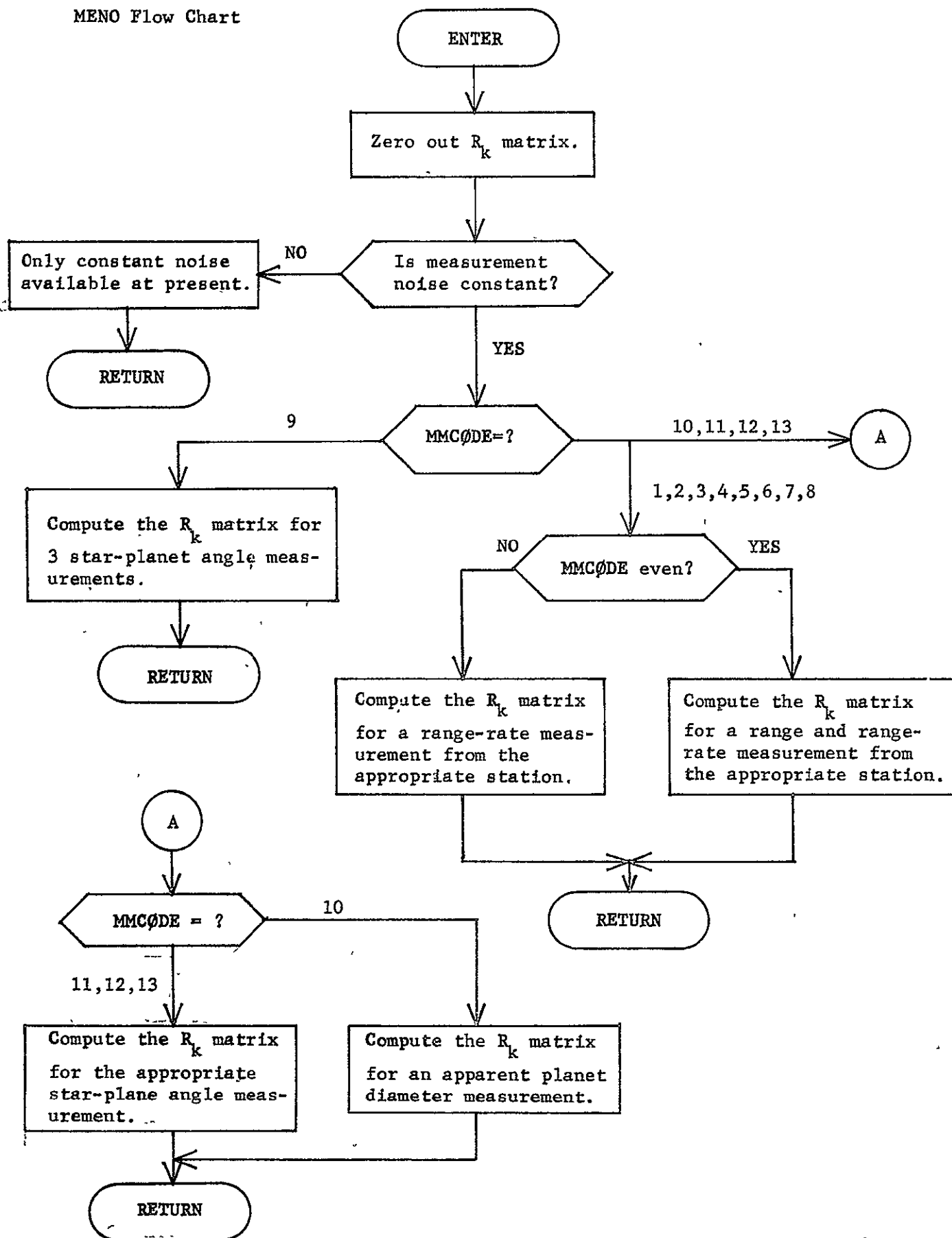
If ICØDE \neq 0 the actual measurement noise covariance matrix

$$R_k' = E \begin{bmatrix} \eta_k' & \eta_k'^T \end{bmatrix}$$

where η_k' is the actual measurement noise, is computed instead. In this case subroutine MENØ selects the appropriate actual measurement noise variances from the vector GMNCN to construct R_k' .

The accompanying flow chart indicates the computational flow for computing R_k . An identical procedure is used to compute R_k' .

MENO Flow Chart



SUBROUTINE MENOS

PURPOSE: COMPUTE ASSUMED AND ACTUAL MEASUREMENT NOISE COVARIANCE
MATRICES IN THE SIMULATION PROGRAM

CALLING SEQUENCE: CALL MENOS(MMCODE,ICODE)

ARGUMENT: ICODE I INTERNAL CODE USED TO DISTINGUISH BETWEEN
THE TWO ALTERNATIVES LISTED ABOVE

MMCODE I MEASUREMENT MODEL CODE

SUBROUTINES SUPPORTED: SIMULL

COMMON COMPUTED/USED: R

COMMON COMPUTED: AR

COMMON USED: AVARM IAMNF IMNF MNCN ZERO

MENOS Analysis

The linearized observation equation employed by the navigation process is given by

$$\delta y_k = H_k^A \delta x_k^A + \eta_k$$

where δy_k is the measurement deviation from the nominal measurement, H_k^A is the augmented observation matrix, δx_k^A is the augmented state deviation from the nominal augmented state, and η_k is the assumed measurement noise.

The actual measurement y_k^a is given by

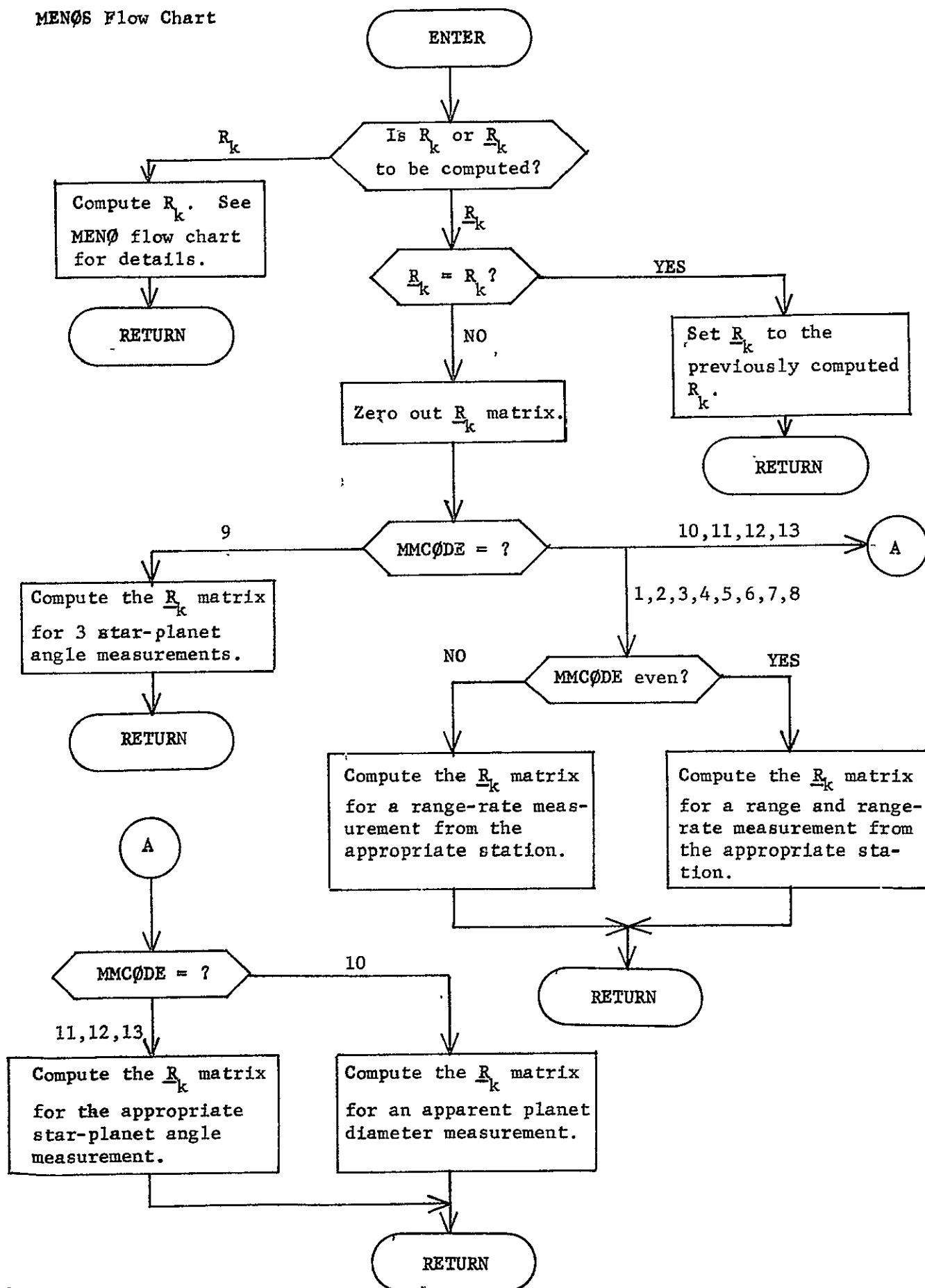
$$y_k^a = \underline{y}_k + b_k + v_k$$

where \underline{y}_k is the ideal measurement, which would be made in the absence of instrumentation errors, b_k is the actual measurement bias, and v_k represents the actual measurement noise.

Subroutine MENOS performs two functions.. It's first function, which is identical to that of subroutine MENO, is to compute the measurement noise covariance matrix R_k which describes the statistics of noise η_k . The constant variances for the assumed measurement noises associated with all available measurement devices are stored in the vector MNCN. Subroutine MENOS selects the appropriate elements from this vector to construct the measurement noise covariance matrix R_k .

The second function of MENOS is to compute the measurement noise covariance matrix R_k which describes the statistics of the actual noise v_k . The constant variances for the actual measurement noises associated with all available measurement devices are stored in the vector AVARM. Subroutine AVARM selects the appropriate elements from this vector to construct the measurement noise covariance matrix R_k .

MENOS Flow Chart



SUBROUTINE MINIQ

PURPOSE: TO COMPUTE EXECUTION ERROR COVARIANCE MATRIX FOR
MINI-PROBE RELEASE

CALLING SEQUENCE: CALL MINIQ(ISACT)

ARGUMENTS: ISACT I ISACT=0 IF OPERATING IN ERRAN
 =1 IF OPERATING IN SIMUL

SUBROUTINES SUPPORTED: PROBE

LOCAL SYMBOLS: ZERO 0.0

 CAPA MINI-PROBE ROLL RELEASE ANGLE

 CCAPA COS(CAPA)

 DTR RADIAN VALUE OF ONE DEGREE

 DVDA PARTIAL OF V WRT ALFA

 DVDD PARTIAL OF V WRT DELT

 DVDL PARTIAL OF V WRT ROOM LENGTH

 DVDP PARTIAL OF V WRT PHI

 DVDW PARTIAL OF V WRT OMEGA

 FACTR 2*PI/3 RADIANS

 SCAPA SIN(CAPA)

 U INTERMEDIATE VECTOR

 V INTERMEDIATE VECTOR

 XX INTERMEDIATE VECTOR

 ZZ INTERMEDIATE VECTOR

COMMON COMPUTED/USED: ADV QT

COMMON USED:	ABW	ALFA	DA	DD	DELT
	DL	DP	DW	EE	XPHI
	YYL				

MINIQ Analysis

Subroutine MINIQ computes the execution error covariance matrix and the actual execution error associated with the spin-release of a miniprobe. The actual execution error is computed only when MINIQ is used in the simulation program SIMUL.

The velocity increment imparted to the i th probe at release is given by

$$\Delta \vec{V}^i = \vec{\omega} \times \vec{r}^i \quad (1)$$

where $\vec{\omega}$ is the spin vector and \vec{r}^i denotes the position of the i th probe relative to the primary vehicle. Referred to the $\hat{u} \hat{v} \hat{h}$ coordinate system, which is defined in subroutine TPRTRG, equation (1) becomes

$$\Delta \vec{V}^i = l \omega \cos \left\{ \left[\phi + (i-1) \frac{2\pi}{3} \right] \hat{u} + \sin \left[\phi + (i-1) \frac{2\pi}{3} \right] \hat{v} \right\} \quad (2)$$

where $\left[\phi + (i-1) \frac{2\pi}{3} \right]$ is the roll release angle of the i th probe, and \hat{u} and \hat{v} are unit vectors.

Define $\vec{p} = (\omega, l, \alpha, \delta, \phi)$ as the release execution parameter vector, where ω is the spin rate magnitude, l is the boom length, α is the right ascension of the spin axis, δ is the declination of the spin axis, and ϕ is the roll release angle. Then the release execution error can be written as

$$\delta \Delta \vec{V}^i = \frac{\partial \Delta \vec{V}^i}{\partial \vec{p}} \delta \vec{p} \quad (3)$$

where $\delta \vec{p}$ represents the error in the release parameter vector.

The j th component of $\delta \Delta \vec{V}^i$ is given by

$$\delta \Delta V_j^i = \sum_{m=1}^5 \frac{\partial \Delta V_j^i}{\partial p_m} \delta p_m \quad (4)$$

The execution error covariance matrix is defined as

$$\tilde{Q}^i = E \left[\delta \Delta \vec{V}^i \cdot \delta \Delta \vec{V}^{iT} \right] \quad (5)$$

and the element \tilde{Q}_{jk}^i of matrix \tilde{Q}^i is given by

$$\tilde{Q}_{jk}^i = E \left[\sum_{m=1}^5 \frac{\partial \Delta V_j^i}{\partial p_m} \delta p_m \cdot \sum_{n=1}^5 \frac{\partial \Delta V_k^i}{\partial p_n} \delta p_n \right] \quad (6)$$

Assuming $E \left[\delta p_m \delta p_n \right] = 0$ for $m \neq n$, equation (6) reduces to

$$\tilde{Q}_{jk}^i = \sum_{m=1}^5 \frac{\partial \Delta V_j^i}{\partial p_m} \cdot \frac{\partial \Delta V_k^i}{\partial p_m} \cdot E \left[\delta p_m^2 \right] \quad (7)$$

The partial derivatives required for evaluation of equations (4) and (7) are summarized as:

$$\frac{\partial \Delta \vec{V}^i}{\partial \omega} = \ell \left\{ \cos \left[\phi + (i-1) \frac{2\pi}{3} \right] \hat{u} + \sin \left[\phi + (i-1) \frac{2\pi}{3} \right] \hat{v} \right\} \quad (8)$$

$$\frac{\partial \Delta \vec{V}^i}{\partial \ell} = \omega \left\{ \cos \left[\phi + (i-1) \frac{2\pi}{3} \right] \hat{u} + \sin \left[\phi + (i-1) \frac{2\pi}{3} \right] \hat{v} \right\} \quad (9)$$

$$\frac{\partial \Delta \vec{V}^i}{\partial \alpha} = \ell \omega \left\{ \cos \left[\phi + (i-1) \frac{2\pi}{3} \right] \frac{\partial \hat{u}}{\partial \alpha} + \sin \left[\phi + (i-1) \frac{2\pi}{3} \right] \frac{\partial \hat{v}}{\partial \alpha} \right\} \quad (10)$$

$$\frac{\partial \Delta \vec{V}^i}{\partial \delta} = \ell \omega \left\{ \cos \left[\phi + (i-1) \frac{2\pi}{3} \right] \frac{\partial \hat{u}}{\partial \delta} + \sin \left[\phi + (i-1) \frac{2\pi}{3} \right] \frac{\partial \hat{v}}{\partial \delta} \right\} \quad (11)$$

$$\frac{\partial \Delta \vec{V}^i}{\partial \phi} = \ell \omega \left\{ -\sin \left[\phi + (i-1) \frac{2\pi}{3} \right] \hat{u} + \cos \left[\phi + (i-1) \frac{2\pi}{3} \right] \hat{v} \right\} \quad (12)$$

where

$$\hat{u} = (\sin \alpha, -\cos \alpha, 0) \quad (13)$$

$$\hat{v} = (\sin \delta \cos \alpha, \sin \delta \sin \alpha, -\cos \delta) \quad (14)$$

$$\frac{\partial \hat{u}}{\partial \alpha} = (\cos \alpha, \sin \alpha, 0) \quad (15)$$

$$\frac{\partial \hat{v}}{\partial \alpha} = (-\sin \delta \sin \alpha, \sin \delta \cos \alpha, 0) \quad (16)$$

$$\frac{\partial \hat{u}}{\partial \delta} = (0, 0, 0) \quad (17)$$

$$\frac{\partial \hat{v}}{\partial \delta} = (\cos \delta, \cos \alpha, \cos \delta \sin \alpha, \sin \delta) \quad (18)$$

when referred to the ecliptic coordinate system.

SUBROUTINE MOMENT

PURPOSE: TO CONVERT AN ARBITRARY NON-SQUARE 2ND MOMENT MATRIX TO THE ASSOCIATED CORRELATION MATRIX PARTITION AND PRINT IT. ALSO COMPUTE AND PRINT EIGENVALUES, EIGENVECTORS, AND HYPERELLIPSOIDS

CALLING SEQUENCE: CALL MOMENT(N1,N2,EXYT,EX,EY,CORW,CORW1,ABL,I1,I2,IFLAG,IF2)

ARGUMENTS: N1 I NUMBER OF ROWS IN 2ND MOMENT MATRIX
 N2 I NUMBER OF COLS IN 2ND MOMENT MATRIX
 EXYT I N1 BY N2 2ND MOMENT MATRIX OF X AND Y
 EX I N1 VECTOR MEAN OF X
 EY I N2 VECTOR MEAN OF Y
 CORW I 2ND MOMENT MATRIX CORRESPONDING TO VECTOR X OF DIMENSION N1
 CORW1 I 2ND MOMENT MATRIX CORRESPONDING TO VECTOR Y OF DIMENSION N2
 ABL I VECTOR OF ROW LABELS CORRESPONDING TO CORW1
 I1 I ROW INDEX MAXIMUM
 I2 I COL INDEX MAXIMUM
 IFLAG I =0 DO NOT COMPUTE EIGENVECTORS, ETC.
 IF2 I =0 DO NOT COMPUTE STD. DEV.

SUBROUTINES SUPPORTED: GPRINT GENGID

SUBROUTINES REQUIRED: EIGHTY

LOCAL SYMBOLS: OUT INTERMEDIATE ARRAY
 PEIG INTERMEDIATE VECTOR
 ROW INTERMEDIATE VECTOR
 SQP INTERMEDIATE VECTOR
 SQP1 INTERMEDIATE VECTOR
 VEIG INTERMEDIATE VECTOR

	Z7	INTERMEDIATE VARIABLE	
COMMON USED:		FOP	FOV

MØMENT Analysis

Subroutine MØMENT transforms an arbitrary 2nd moment matrix $E[xy^T]$ into a correlation matrix and, if $x = y$, into a vector of standard deviations. The transformation consists of two steps:

- 1) Transform $E[xy^T]$ into the covariance matrix

$$\text{cov } (x,y) = E[xy^T] - E[x] \cdot E[y^T];$$

- 2) Transform $\text{cov } (x,y)$ into the correlation matrix having correlation coefficients

$$\rho_{ij} = \frac{\sigma_{ij}}{\sigma_i \sigma_j} \quad i \neq j$$

where

$$\sigma_{ij} = E[x_i y_j]$$

$$\sigma_i = E[x_i^2]^{1/2}$$

$$\sigma_j = E[y_j^2]^{1/2}$$

Subroutine MØMENT writes out the correlation matrix and, if they exist, the standard deviations. Subroutine MØMENT can also compute and write out the eigenvalues, eigenvectors, and hyperellipsoid of $\text{cov } (x,y)$ if $x = y$.

SUBROUTINE MPPROP

PURPOSE: TO GENERATE A TIME HISTORY OF MAIN PROBE TRAJECTORY

SUBROUTINES SUPPORTED: GIDANS

SUBROUTINES REQUIRED: VMP

LOCAL SYMBOLS: DELTM MAXIMUM DURATION OF PROPAGATION IN DAYS

DELTPS SAVED VALUE OF VMP VARIABLE DELTP
SPECIFYING TIME INTERVAL IN DAYS BETWEEN
SUCCESSIVE TRAJECTORY STATUS PRINTOUTS

ICLS SAVED VALUE OF VMP CONDITION CODE ICL
INDICATING WHETHER OR NOT CLOSEST APPROACH
HAS BEEN REACHED

ICL2S SAVED VALUE OF VMP TERMINATION CODE ICL2
SPECIFYING WHETHER OR NOT TRAJECTORY IS TO
BE STOPPED AT CLOSEST APPROACH

IEPS SAVED VALUE OF VMP FLAG IEPHEM SPECIFYING
WHETHER OR NOT ORB IS TO BE CALLED BEFORE
CALLING EPHEM

INPRS SAVED VALUE OF VMP VARIABLE INPR
SPECIFYING NUMBER OF INTEGRATION STEPS
BETWEEN SUCCESSIVE TRAJECTORY STATUS
PRINTOUTS

IPRNTS SAVED VALUE OF VMP FLAG IPRINT
SPECIFYING WHETHER OR NOT TRAJECTORY
INFORMATION IS TO BE PRINTED

ISPHS SAVED VALUE OF VMP CONDITION CODE ISPH
SPECIFYING WHETHER OR NOT TRAJECTORY IS
TO BE STOPPED AT SOI

TRTHS SAVED VALUE OF VMP VARIABLE TRTM
INDICATING ELAPSED TIME IN DAYS FROM
START OF TRAJECTORY

XSF FINAL HELIOCENTRIC ECLIPTIC STATE VECTOR
OF MAIN PROBE AFTER PROPAGATION IN KM AND
KM/SEC

COMMON COMPUTED/USED: DELTP FIVE ICL ICL2 IEPHEM
IPCS IPRINT ISPH KUR NINETY
TRTM

COMMON COMPUTED: DELTP ICL ICL2 IEPHEM INCMT
IPCSP IPRINT ISPH RPSP TRTM

MPPRØP Analysis

MPPRØP serves the single purpose of providing a time history of the targeted main probe trajectory from release to the appropriate stopping condition. The process of providing such a time history is classified as a separate type of guidance event in the GIDANS execution logic. Although the main probe propagation event is currently only applied in the NØMNAL program to propagate the main probe once it begins to deviate from the bus trajectory due to some guidance maneuver on the latter, it could be used to treat any branched trajectory.

On being called by GIDANS, MPPRØP first prints out the title "Main Probe Propagation Event" followed by the heading "Main Probe Approach Trajectory." Next it stores the current spacecraft heliocentric ecliptic state and the VMP trajectory condition and instrument flags so that they can be returned to after the event. Then the VMP flags are set for propagating the main probe. Codes are used to stop the trajectory after 90 days of propagation or at closest approach but not at the sphere of influence. The VMP stopping condition of impacting the planet is slightly modified. Rather than using NØMNAL's own value of target-planet radius, MPPRØP transmits to VMP the radius of the probe sphere input by the user and applied throughout probe targeting. The print routine is activated and the print increments are set at 5-day and 100-integration steps. Next VMP is called to propagate and print the trajectory until a stopping condition is reached. Finally, the original spacecraft state and the VMP flags are restored before returning to GIDANS.

SUBROUTINE MULCON

PURPOSE: TO PROPAGATE A SET OF CARTESIAN COORDINATES ALONG A LUNAR MULTI-CONIC TRAJECTORY OVER A SPECIFIED TIME INTERVAL.

CALLING SEQUENCE: CALL MULCON(SEI,TLI,TF,DT,SMF)

ARGUMENTS: SEI(6) I INITIAL SPACECRAFT GEOCENTRIC STATE
 TLI I INITIAL INJECTION TIME (JD EPOCH 1950)
 TF I TIME INTERVAL OF PROPAGATION
 DT I STEP SIZE USED IN MULTICONIC PROPAGATION
 SMF(6) O FINAL SPACECRAFT SELENOCENTRIC STATE

SUBROUTINES SUPPORTED: MULTAR

SUBROUTINES REQUIRED: CPROP EPHEM ORB

LOCAL SYMBOLS: ALNGTH CONVERTS KILOMETERS TO ASTRONOMICAL UNITS
 A PERTURBING ACCELERATION VECTOR OF THE MOON OVER THE ITERATION INTERVAL
 COSF COSINE OF TRUE ANOMALY ON SELENOCENTRIC CONIC AT END OF ITERATION INTERVAL
 DF FINAL TIME USED IN ITERATION INTERVAL
 DI INITIAL TIME USED IN ITERATION INTERVAL
 EMM MAGNITUDE OF FIRST THREE ELEMENTS OF EM
 EM GEOCENTRIC ECLIPTIC STATE OF MOON
 IDONE STOPPING CONDITION INDICATOR
 =0 PROPAGATION CONTINUES
 =1 STOPPING CONDITION REACHED
 I INDEX
 TIM JULIAN DATE OF FINAL TIME ON THE ITERATION INTERVAL
 TM CONVERTS SECONDS TO DAYS
 W SPACECRAFT VELOCITY VECTOR WITH RESPECT TO EARTH AND/OR MOON BEFORE AND AFTER LUNAR PERTURBATIONS AT DI AND DF

XM	MAGNITUDE OF THE X VECTOR
X	GEOCENTRIC POSITION OF SPACECRAFT AT DI AND GEOCENTRIC POSITION OF MOON AT DF
Y	GEOCENTRIC VELOCITY OF SPACECRAFT AT DI AND GEOCENTRIC VELOCITY OF MOON AT DF
Z	SPACECRAFT POSITION VECTOR WITH RESPECT TO EARTH AND/OR MOON AT DI AND DF BEFORE AND AFTER LUNAR PERTURBATIONS

COMMON COMPUTED: NO

COMMON USED: EMU TMU TWO XP ZERO

MULCON Analysis

The equations of motion of a spacecraft traveling under the influence of the earth and moon may be written

$$\ddot{\vec{r}}_E = -\frac{\mu_E \vec{r}_E}{r_E^3} - \frac{\mu_M \vec{r}_M}{r_M^3} - \frac{\mu_M \vec{R}_{EM}}{r_{EM}^3} \quad (1)$$

where \vec{r}_E , \vec{r}_M , \vec{R}_{EM} are the position vectors of the spacecraft-to-earth, the spacecraft-to-moon, and the moon-to-earth respectively and μ_E , μ_M are the gravitational constants of the earth and moon respectively.

The multi-conic approximation of the solution to (1) proceeds as follows. Let $\vec{r}_{E,k}$, $\vec{v}_{E,k}$ be the geocentric state at some time t_k . This state is propagated by conic formulae to obtain an estimate of the geocentric state at time $t_{k+1} = t_k + \Delta t$ given by $\vec{r}_{E,k+1}$, $\vec{v}_{E,k+1}$.

To account for the third term perturbations, the state of the moon relative to the earth at the two timepoints is computed, denoted by $(\vec{R}_{EM,k}, \vec{v}_{EM,k})$ and $(\vec{R}_{EM,k+1}, \vec{v}_{EM,k+1})$. The average value of this acceleration is then determined from

$$\vec{A} = -\frac{\mu_M}{2} \left[\frac{\vec{R}_{EM,k}}{R_{EM,k}^3} + \frac{\vec{R}_{EM,k+1}}{R_{EM,k+1}^3} \right] \quad (2)$$

The corrected geocentric state is then given by

$$\begin{aligned} \vec{r}_{E,k+1}'' &= \vec{r}_{E,k+1}' + \frac{1}{2} \vec{A} (\Delta t)^2 \\ \vec{v}_{E,k+1}'' &= \vec{v}_{E,k+1}' + \vec{A} \Delta t \end{aligned} \quad (3)$$

The effect of the direct lunar perturbations is then added. The state of the spacecraft relative to the moon is first computed

$$\begin{aligned} \vec{r}_{M,k+1}' &= \vec{r}_{E,k+1}'' - \vec{R}_{EM,k+1} \\ \vec{v}_{M,k+1}' &= \vec{v}_{E,k+1}'' - \vec{v}_{EM,k+1} \end{aligned} \quad (4)$$

This state is then propagated linearly backwards in time over the time interval Δt to obtain

$$\begin{aligned}\vec{r}_{M,k}' &= \vec{r}_{M,k+1}' - \vec{v}_{M,k+1}' \Delta t \\ \vec{v}_{M,k}' &= \vec{v}_{M,k+1}'\end{aligned}\tag{5}$$

This state is now propagated forward in a selenocentric conic to obtain a final state relative to the moon $(\vec{r}_{M,k+1}', \vec{v}_{M,k+1}')$. The geocentric state of the spacecraft at time t_{k+1} after considering all terms of (1) is then given by

$$\begin{aligned}\vec{r}_{E,k+1} &= \vec{r}_{M,k+1}' + \vec{R}_{EM,k+1} \\ \vec{v}_{E,k+1} &= \vec{v}_{M,k+1}' + \vec{V}_{EM,k+1}\end{aligned}\tag{6}$$

This completes one cycle of the multi-conic propagation.

The multi-conic propagation proceeds until an input final time is reached or until the selenocentric conic passes through pericynthion.

Reference: Byrnes, D. V. and Hooper, H. L., Multi-Conic: A Fast and Accurate Method of Computing Space Flight Trajectories, AAS/AIAA Astrodynamics Conference, Santa Barbara, Cal., 1970, AIAA Paper 70-1062.

SUBROUTINE MULTAR

PURPOSE: TO CALCULATE THE TRANSLUNAR INJECTION CONDITIONS FROM TARGETED PATCHED-CONIC CONDITIONS AND CALLS VMP TO PERFORM THE NOMINAL TRAJECTORY NEEDED BY ITERAT.

CALLING SEQUENCE: CALL MULTAR

SUBROUTINES SUPPORTED: LUNA

SUBROUTINES REQUIRED: MULCON CAREL ELCAR EPHEM IMPACT
MATIN ORB PECEQ TIME

LOCAL SYMBOLS: AE SEMI-MAJOR AXIS OF THE EARTH-ECLIPTIC,
TARGETED PATCHED-CONIC TRAJECTORY

ATARN NOMINAL VALUES OF THE TARGET VARIABLES

ATAR DESIRED VALUES OF THE TARGET VARIABLES

ATOL TOLERANCES OF TARGET VARIABLES

BCOR MAXIMUM STEPS ALLOWED IN ITERATIVE
CORRECTION OF CONTROL VARIABLES

BJ ZERO TRUE ANOMALY USED TO DEFINE PERIGEE
OF THE TARGETED PATCHED-CONIC TRAJECTORY

BSTEP MULTI-CONIC STEP SIZE (HOURS)

CHI SENSITIVITY MATRIX RELATING PERTURBATIONS
IN CONTROL VARIABLES TO CHANGES IN TARGET
VARIABLES

DELP VALUE USED TO PERTURB TLI FOR CONSTRUCTION
OF CHI

DELTH NOMINAL TIME FOR PROPAGATION

DELT INTEGRATION TIME TO BE USED, AND TIME
ACTUALLY USED, IN THE MULTI-CONIC
PROPAGATION

DELV VALUE USED TO PERTURB VELOCITY COMPONENTS
OF RT FOR CONSTRUCTION OF CHI

DV CORRECTION ACTUALLY ADDED TO CONTROL
VARIABLES

ECEQP TRANSFORMATION MATRIX FROM EARTH ECLIPTIC
TO LUNAR EQUATORIAL

EECEQ TRANSFORMATION MATRIX FROM EARTH ECLIPTIC

TO EARTH EQUATORIAL

EE	ECCENTRICITY OF THE EARTH-ECLIPTIC, TARGETED PATCHED-CONIC TRAJECTORY
ERR	ITERATE ERRORS IN TARGET CONDITIONS
FAC	INTERMEDIATE VARIABLE USED TO CHECK FOR MAXIMUM STEP
FMAG	INTERMEDIATE VARIABLE USED TO COMPUTE PHIA, PHIB, PHIC
HYT	HYPERBOLIC TIME TO LUNAR PERIAPSIS (DAYS)
IDATE	CALENDAR DATE OF INJECTION
IST	INDICATOR FOR CONTROL VARIABLE PERTURBATION
IT	ITERATIONS COUNTER
I	INDEX
JERTH	INDEX OF EARTH IN F-ARRAY
JMOON	INDEX OF MOON IN F-ARRAY
J	INDEX
K	INDEX
MITS	MAXIMUM NUMBER OF ITERATIONS ALLOWED
NDEX	SAME AS JERTH
PERMN	MINIMUM PERTURBATION OF CONTROL VARIABLES FOR CONSTRUCTION OF CHI
PERMX	MAXIMUM PERTURBATION OF CONTROL VARIABLES FOR CONSTRUCTION OF CHI
PERT	PERTURBATION VALUES USED TO CONSTRUCT CHI
PHIA	TRANSFORMATION MATRIX FROM RTN TO EC SYSTEM AT TLI
PHIB	TRANSFORMATION MATRIX FROM RTN TO EC SYSTEM AT PERTURBED TLI
PHIC	PRODUCT OF PHIB AND TRANSPOSE OF PHIA
PPE	DUMMY VARIABLE FOR CALL TO CAREL

PSI	TARGET MATRIX (INVERSE OF CHI) RELATING PERTURBATIONS IN TARGET VARIABLES TO CHANGES IN CONTROL VARIABLES
PTAR	PERTURBED TARGET VALUES
PV	PREDICTED CORRECTIONS TO CONTROL VARIABLES
QQE	DUMMY VARIABLE FOR CALL TO CAREL
REMAG	DUMMY VARIABLE FOR CALL TO ELCAR
REPET	MINIMUM ALLOWABLE INJECTION TIME DIFFERENCE IN KTH AND K+2 ITERATIONS TO AVOID REPETITION-TRAP CORRECTION
RM	MAGNITUDE OF THE SCM POSITION VECTOR
RSE	INJECTION STATE IN EARTH EQUATORIAL SYSTEM AT TLI
RS	ROTATED INJECTION STATE FOR TIME DIFFERENTIAL
RT	INJECTION STATE USED IN PERTURBED MULTI- CONIC PROPAGATIONS
SCM	FINAL STATE IN LUNAR ECLIPTIC SYSTEM ON THE MULTI-CONIC
SEC	SECONDS OF CALENDAR DATE OF TLI
STEP	MULTI-CONIC STEP SIZE (SECONDS)
STLI	ORIGINAL VALUE OF TLI, RESTORED FOR SUCESSIVE ITERATIONS
TAE	TRUE ANOMALY OF EARTH-ECLIPTIC TARGETED PATCHED-CONIC TRAJECTORY
TBR	DUMMY VARIABLE FOR CALL TO IMPACT
TBT	DUMMY VARIABLE FOR CALL TO IMPACT
TFP	TIME OF FLIGHT FROM PERIGEE OF THE EARTH-ECLIPTIC, TARGETED PATCHED-CONIC TRAJECTORY
TIMM1	INJECTION DATE ON K-1 ITERATION
TIMM2	INJECTION DATE ON K-2 ITERATION

TLI INJECTION JULIAN DATE
 TTP DUMMY VARIABLE FOR CALL TO ELCAR
 VEMAG DUMMY VARIABLE FOR CALL TO ELCAR
 VX MAGNITUDE OF THE SCM VELOCITY VECTOR
 WE ARGUMENT OF PERIAPSIS OF THE
 EARTH-ECLIPTIC, TARGETED PATCHED-CONIC
 TRAJECTORY
 WHE DUMMY ARGUMENT FOR CALL TO CAREL
 XIE INCLINATION OF THE EARTH-ECLIPTIC,
 TARGETED PATCHED-CONIC TRAJECTORY
 XNE LONGITUDE OF ASCENDING NODE OF THE
 EARTH-ECLIPTIC, TARGETED PATCHED-CONIC
 TRAJECTORY

COMMON COMPUTED/USED:

NO RI

COMMON COMPUTED:

ICCOORD RIN TIN

COMMON USED:

ALNGTH	CAI	EMU	F	HALF
KUR	NBOD	NB	ONE	RCC
SMA	TAR	TCA	TEN	TMU
TH	TSI	TWO	ZERO	

MULTAR Analysis

Let the earth equatorial state of the probe at the LSI as computed from the patched conic targeting be denoted $\vec{r}_{LS}, \vec{v}_{LS}$. Subroutine CAREL is called to compute the conic elements and conic time from perigee Δt based on the geocentric conic. The time of injection is then computed as

$$t_{TLI} = t_{SI} - \Delta t \quad (1)$$

The position and velocity of the probe at t_{TLI} is given by the state along the conic at perigee (true anomaly of zero) and determined by ELCAR to be $\vec{r}_{TLI}, \vec{v}_{TLI}$. If Φ_{ECEQ} is the transformation matrix from the EC (earth ecliptic) to the EQ (earth equatorial) system, then the patched conic injection state in EC coordinates is

$$\begin{aligned} \vec{r}_I &= \Phi_{ECEQ}^T \vec{r}_{TLI} \\ \vec{v}_I &= \Phi_{ECEQ}^T \vec{v}_{TLI} \end{aligned} \quad (2)$$

Since the earth is revolving about the E-M barycenter in time, the EC injection state must be rotated if an earlier or later injection time is to be used. The necessary rotation matrix may be easily computed through the introduction of the R-T-W coordinate system. Let the state of the earth at some time t_k in BC' (barycentric ecliptic) coordinates be denoted \vec{R}_k, \vec{V}_k . Construct the $\hat{R}-\hat{T}-\hat{W}$ system at that point as

$$\hat{R}_k = \frac{\vec{R}_k}{R_k}, \quad \hat{W}_k = \frac{\vec{R}_k \times \vec{V}_k}{|\vec{R}_k \times \vec{V}_k|}, \quad \hat{T}_k = \hat{W}_k \times \hat{R}_k \quad (3)$$

The transformation matrix from the $\hat{R}_k-\hat{T}_k-\hat{W}_k$ system to the ecliptic system to the ecliptic system is then given by

$$\Phi_k = \begin{bmatrix} \hat{R}_k & \hat{T}_k & \hat{W}_k \end{bmatrix} \quad (4)$$

At a time t_{k+1} the state of the earth in BC coordinates is given by $\vec{R}_{k+1}, \vec{V}_{k+1}$ and the transformation from the $\hat{R}_{k+1}-\hat{T}_{k+1}-\hat{W}_{k+1}$ system to ecliptic coordinates is given by Φ_{k+1} in accordance with (4). Injection

states at times t_k and t_{k+1} will be called "equivalent" if they are identical when expressed in the pertinent $\hat{R}-\hat{T}-\hat{W}$ system. Therefore if (\vec{r}_k, \vec{v}_k) is the injection state in EC coordinates at time t_k , the equivalent state in EC coordinates at t_{k+1} is given by

$$\begin{bmatrix} \vec{r}_{k+1} \\ \vec{v}_{k+1} \end{bmatrix} = \begin{bmatrix} \psi_{k+1,k} & 0 \\ 0 & \psi_{k+1,k} \end{bmatrix} \begin{bmatrix} \vec{r}_k \\ \vec{v}_k \end{bmatrix} \quad (5)$$

where the rotation matrix ψ is defined by

$$\psi_{k+1,k} = \phi_{k+1} \phi_k^T \quad (6)$$

The targeting algorithm used by MULTAR may now be described. Let the injection state in EC coordinates on the k -th iteration be denoted $(t_k, \vec{r}_k, \vec{v}_k)$. This state is propagated forward using the multi-conic propagator MULCON to determine a final state \vec{r}_M, \vec{v}_M to the moon in ecliptic coordinates. IMPACT is then called to compute the $B \cdot T_k$, $B \cdot R_k$ and $t_{CA,k}$ actually achieved on the trajectory and the target values of $B^* \cdot T_k$, $B^* \cdot R_k$ required to satisfy the i_{CA} and r_{CA} constraints. The semi-major axis a_k of the k -th iterate is computed from the conic formula

$$a = r_M \left(2 - \frac{r_M v_M^2}{\mu_M} \right)^{-1} \quad (7)$$

Errors in the four target conditions

$$\Delta \tau = \begin{bmatrix} \Delta a \\ \Delta B \cdot T \\ \Delta B \cdot R \\ \Delta t_{CA} \end{bmatrix} = \begin{bmatrix} a - a^* \\ B \cdot T_k - B^* \cdot T_k \\ B \cdot R_k - B^* \cdot R_k \\ t_{CA,k} - t_{CA}^* \end{bmatrix} \quad (8)$$

if the error in each parameter is less than the allowable tolerance, the process stops.

If convergence has not been achieved a Newton-Raphson iteration is entered. The four controls are \vec{v}_{k_x} , \vec{v}_{k_y} , \vec{v}_{k_z} , and t_k . For the velocity components a perturbation Δv is added to the pertinent component while the rest of the injection state is held constant before propagating with the multi-conic. For the time perturbation, the rotation matrix ψ_Δ corresponding to the perturbed time $t_k + \Delta t$ (6) is first computed. The injection state used in the perturbed propagation for time is then $[t_k + \Delta t, \psi_\Delta \vec{r}_k, \psi_\Delta \vec{v}_k]$. A sensitivity matrix is computed using the results of the numerical differencing:

$$X = \begin{bmatrix} \frac{\Delta a_x}{\Delta v_x} & \frac{\Delta a_y}{\Delta v_y} & \dots & \\ \frac{\Delta B T_x}{\Delta v_x} & \cdot & & \\ \frac{\Delta B R_x}{\Delta v_x} & \cdot & & \\ \frac{\Delta t_{CA_x}}{\Delta v_x} & & & \frac{\Delta t_{CA_c}}{\Delta t} \end{bmatrix} \quad (9)$$

where in the term $\frac{\Delta \alpha_\beta}{\Delta \beta}$, $\Delta \alpha_\beta$ is the change in the α target parameter produced by the variation of the β control component and $\Delta \beta$ is the change in the β control component. The $k+1$ iterate controls are then given by

$$\Delta C = \begin{bmatrix} \delta v_x \\ \delta v_y \\ \delta v_z \\ \delta t \end{bmatrix} = X^{-1} \Delta \tau \quad (10)$$

The $k+1$ injection state is then computed by first determining the injection state after rotation due to the change in injection time and then adding the injection velocity corrections

$$\begin{aligned} t_{k+1} &= t_k + \delta t \\ \vec{r}_{k+1} &= \psi_\delta \vec{r}_k \\ \vec{v}_{k+1} &= \psi_\delta \vec{v}_k + \delta \vec{v} \end{aligned} \quad (11)$$

The iteration process is repeated until tolerable errors are met. The converged injection state is then integrated in the virtual mass trajectory.

SUBROUTINE MUND

PURPOSE: TO COMPUTE THE AUGMENTED PORTION OF THE STATE TRANSITION MATRIX WHEN THE GRAVITATIONAL CONSTANT OF THE SUN OR OF THE TARGET PLANET HAS BEEN AUGMENTED TO THE BASIC STATE VECTOR.

CALLING SEQUENCE: CALL MUND(RI,RF,POSS)

ARGUMENT: RF I POSITION AND VELOCITY OF THE VEHICLE AT THE
END OF THE TIME INTERVAL

RI I POSITION AND VELOCITY OF THE VEHICLE AT THE
BEGINNING OF THE TIME INTERVAL

POSS I DISTANCE OF THE VEHICLE FROM THE TARGET
PLANET AT THE INITIAL TIME

SUBROUTINES SUPPORTED: PSIM

SUBROUTINES REQUIRED: NTM

LOCAL SYMBOLS: IC COUNTER FOR VARIABLES AUGMENTED TO STATE
VECTOR

RPER ALTERED POSITION AND VELOCITY OF VEHICLE
AT FINAL TIME

SAVE TEMPORARY STORAGE LOCATION FOR GRAVITA-
TIONAL CONSTANTS OF SUN AND TARGET PLANET

COMMON COMPUTED/USED: IPRINT PMASS

COMMON COMPUTED: TXU TXXS TXW

COMMON USED: ALNGTH DELMUP DELMUS IAUGDC IAUGIN
IAUG NTMC NTP SPHERE TM
IAUGW

MUND Analysis

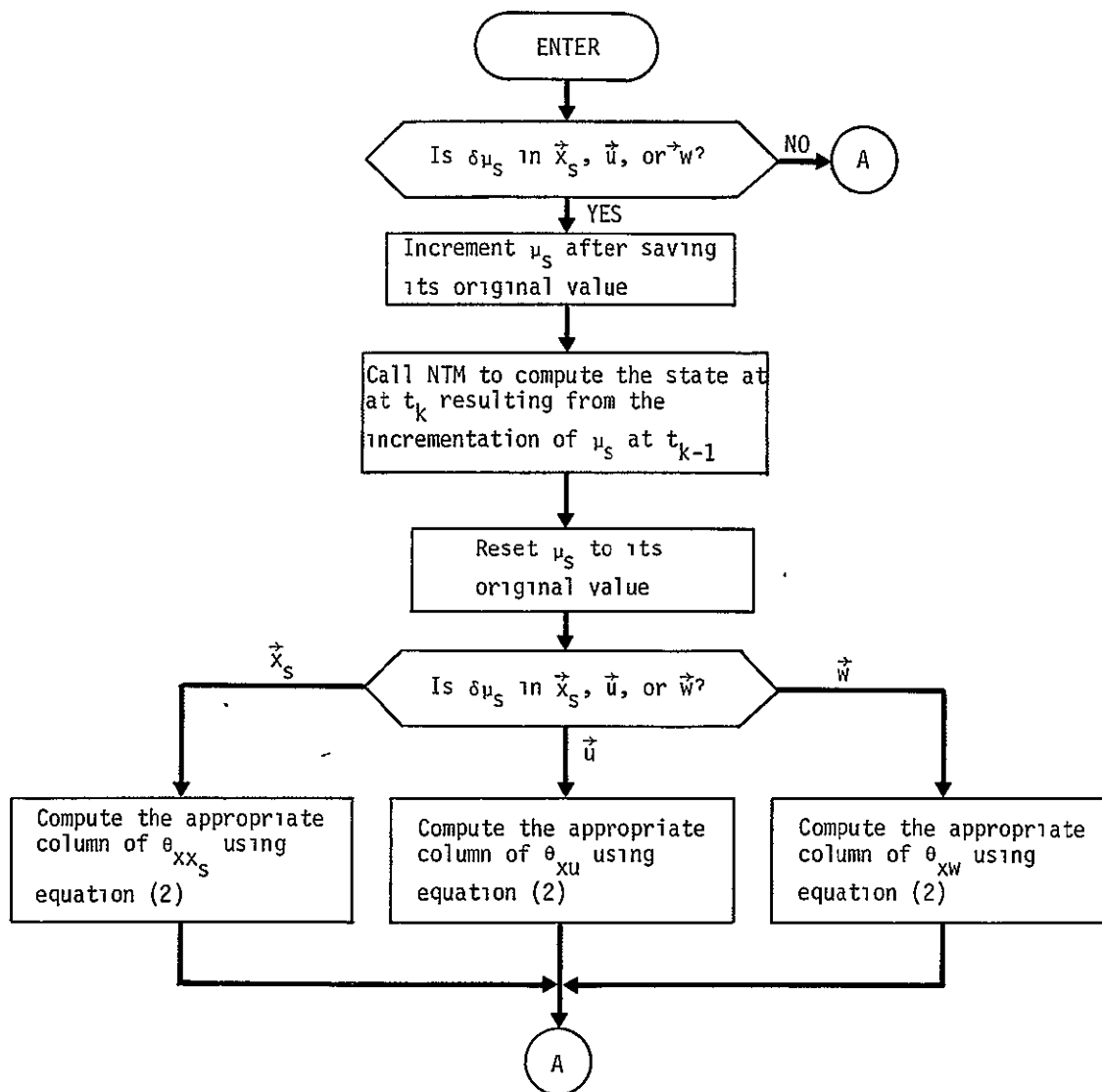
The nonlinear equations of motion of the spacecraft can be written symbolically as

$$\dot{\vec{x}} = \vec{f}(\vec{x}, \vec{\mu}, t) \quad (1)$$

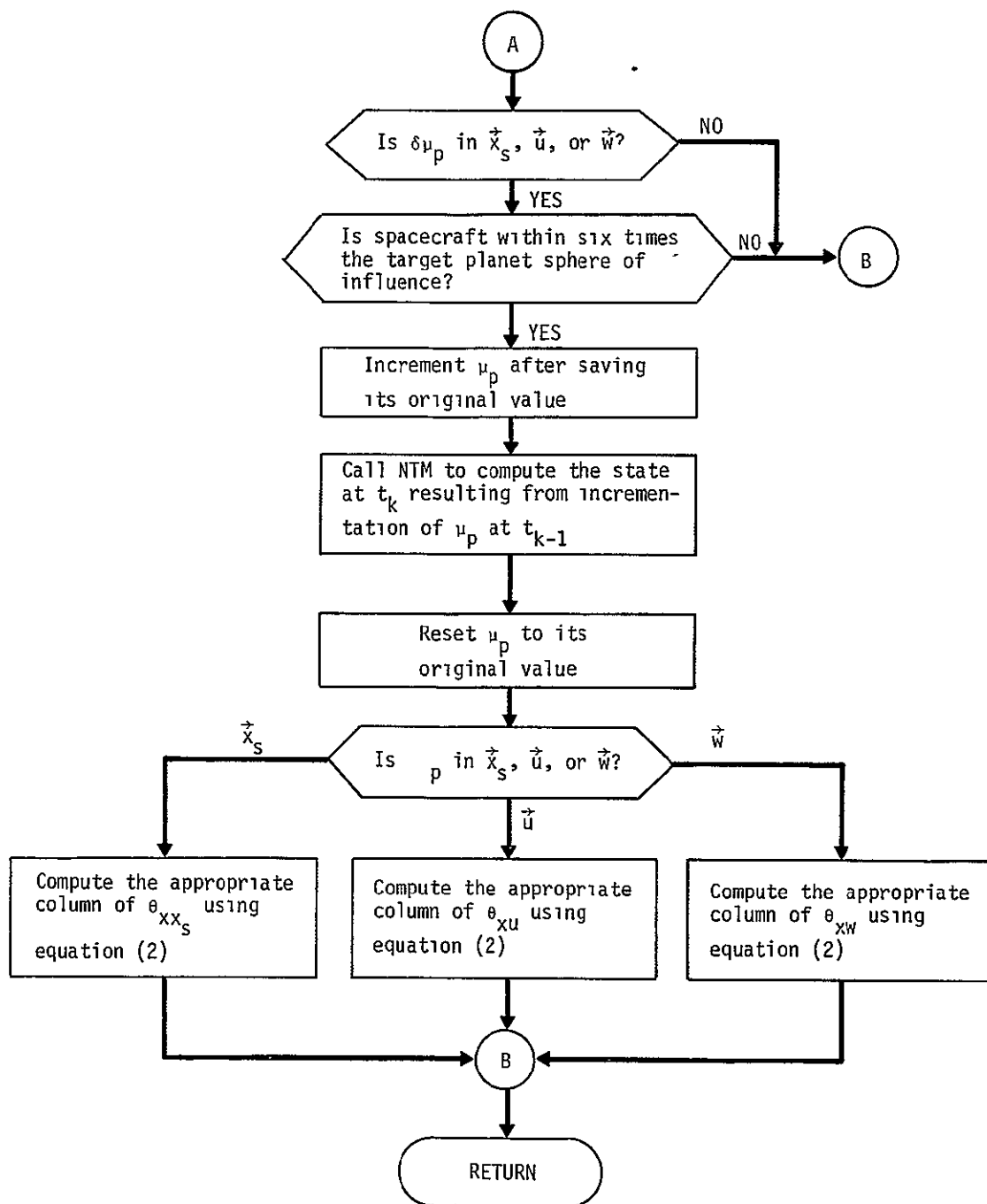
where \vec{x} is the spacecraft position/velocity state and $\vec{\mu}$ is a vector composed of the gravitational constants of the sun and the target planet.

Suppose we wish to use numerical differencing to compute those columns of θ_{xx_s} , θ_{xu} , and θ_{xw} associated with gravitational constant biases included in the augmented state vector over the time interval $[t_{k-1}, t_k]$. Let $\vec{\theta}_j(t_k, t_{k-1})$ represent the column associated with the j -th gravitational constant bias. We assume we have available the nominal states $\vec{x}^*(t_{k-1})$ and $\vec{x}^*(t_k)$, which, of course, were obtained by numerically solving equation (1) using nominal $\vec{\mu}$. To obtain $\vec{\theta}_j(t_k, t_{k-1})$ we increment the j -th gravitational constant bias by the pertinent numerical differencing factor $\Delta\mu_j$ and numerically integrated equation (1) over the interval $[t_{k-1}, t_k]$ to obtain the new spacecraft state $\vec{x}_j(t_k)$, where the j -subscript on the spacecraft state indicates that it was obtained by incrementing the j -th gravitational constant bias. Then

$$\vec{\theta}_j(t_k, t_{k-1}) = \frac{\vec{x}_j(t_k) - \vec{x}^*(t_k)}{\Delta\mu_j} \quad (2)$$



MUND Flow Chart



SUBROUTINE NAVM

PURPOSE: TO PROPAGATE COVARIANCE MATRIX PARTITIONS P, CXXS, CXU, CXV, PS, CXSU, CXSV FROM THE TIME OF THE LAST MEASUREMENT OR EVENT TO THE PRESENT TIME USING A CONSIDER RECURSIVE ALGORITHM.

CALLING SEQUENCE: CALL NAVM(NR,ICODE)

ARGUMENT: ICODE I INTERNAL CODE WHICH DETERMINES IF A MEASUREMENT IS BEING PROCESSED

NR I NUMBER OF ROWS IN THE OBSERVATION MATRIX

SUBROUTINES SUPPORTED: SIMULL SETEVS GUISIM PRESIM PRCBES

SUBROUTINES REQUIRED: MATIN GAIN1 GAIN2

LOCAL SYMBOLS AJ MEASUREMENT RESIDUAL COVARIANCE MATRIX AND ITS INVERSE

AKW INTERMEDIATE ARRAY

DUM INTERMEDIATE VECTOR

PSAVE INTERMEDIATE ARRAY

SUM INTERMEDIATE VARIABLE

SW INTERMEDIATE ARRAY

COMMON COMPUTED/USED:	AK	CXSUP	CXSU	CXSVP	CXSV
	CXUP	CXU	CXVP	CXV	CXXSP
	CXXS	PP	PSP	PS	P
	S				

COMMON USED:	AL	AM	G	H	NCIM1
	NDIM2	NDIM3	ONE	PHI	Q
	R	TXU	TXXS	U0	V0
	ZERO				

NAVM Analysis

The augmented deviation state vector is defined as

$$\vec{x}^A = \left[\vec{x}, \vec{x}_s, \vec{u}, \vec{v} \right]^T$$

where

\vec{x} = position and velocity state (dimension 6)

\vec{x}_s = solve-for parameter state (dimension n_1)

\vec{u} = dynamic consider parameter state (dimension n_2)

\vec{v} = measurement consider parameter state (dimension n_3)

The linearized equations of motion have form

$$\begin{aligned}\dot{\vec{x}} &= F_1 \vec{x} + F_2 \vec{x}_s + F_3 \vec{u} \\ \dot{\vec{x}}_s &= 0 \\ \dot{\vec{u}} &= 0 \\ \dot{\vec{v}} &= 0\end{aligned}$$

and solution

$$\begin{aligned}\vec{x}_{k+1} &= \Phi(k+1, k) \vec{x}_k + \theta_{xx_s}(k+1, k) \vec{x}_{s_k} + \theta_{xu}(k+1, k) \vec{u}_k + \vec{q}_k \\ \vec{x}_{s_{k+1}} &= \vec{x}_{s_k} \\ \vec{u}_{k+1} &= \vec{u}_k \\ \vec{v}_{k+1} &= \vec{v}_k\end{aligned}$$

where dynamic noise \vec{q}_k has been added to the solution of \vec{x}_{k+1} . This solution can be written in augmented form

$$\vec{x}_{k+1}^A = \Phi^A(k+1, k) \vec{x}_k^A + \vec{q}_k^A$$

where the augmented state transition matrix $\Phi^A(k+1, k)$ is defined as

$$\Phi^A(k+1,k) = \begin{bmatrix} \Phi & \theta_{xx_s} & \theta_{xu} & 0 \\ 0 & I_{n_1 \times n_1} & 0 & 0 \\ 0 & 0 & I_{n_2 \times n_2} & 0 \\ 0 & 0 & 0 & I_{n_3 \times n_3} \end{bmatrix}$$

Henceforth state transition matrix partitions will be written without stating the associated interval of time, which will always be assumed to be $[k, k+1]$.

The measurement deviation vector \vec{y} (dimension m) is related to the augmented deviation state vector through the equation

$$\vec{y}_k = H_k^A \vec{x}_k^A + \vec{\eta}_k$$

where the augmented observation matrix is defined as

$$H_k^A = \begin{bmatrix} H_k & M_k & G_k & L_k \end{bmatrix}$$

and $\vec{\eta}_k$ is measurement noise.

The augmented state covariance matrix P_k^A can be written in terms of its partitions as

$$P_k^A = \begin{bmatrix} P_k & C_{xx_s k} & C_{xu k} & C_{xv k} \\ C_{xx_s k}^T & P_{s k} & C_{xs u k} & C_{xs v k} \\ C_{xu k}^T & C_{xs u k}^T & U_o & C_{uv k} \\ C_{xv k}^T & C_{xs v k}^T & C_{uv k}^T & V_o \end{bmatrix}$$

Propagation and update equations for the partitions appearing in the previous equation will be written below. Equations need not be written for the consider parameter covariances U_0 and V_0 since these do not change with time. Also, C_{uv} will be set to zero because of the assumption that no cross-correlation exists between dynamic- and measurement-consider parameters. In the equations below, Q and R represent the covariances of the dynamic and measurement noises, respectively, defined previously. A minus superscript on covariance partitions indicates the covariance partition immediately prior to processing a measurement; a plus superscript, immediately after processing a measurement. If ICØDE indicates that a measurement is not to be processed, the update equations are bypassed. To improve numerical accuracy and avoid nonpositive definite covariance matrices, P^- , P_s^- , P^+ , and P_s^+ , are always symmetrized after their computation.

The propagation equation are given by

$$P_{k+1}^- = \left(\phi P_k^+ + \theta_{xx_s} C_{xx_s k}^{+T} + \theta_{xu} C_{xu_k}^{+T} \right) \phi^T +$$

$$C_{xx_s k+1}^- \theta_{xx_s}^T + C_{xu_{k+1}}^- \theta_{xu}^T + Q_k$$

$$C_{xx_s k+1}^- = \phi C_{xx_s k}^- + \theta_{xx_s} P_{s_k}^+ + \theta_{xu} C_{x_s u_k}^{+T}$$

$$P_{s_{k+1}}^- = P_{s_k}^+$$

$$C_{xu_{k+1}}^- = \phi C_{xu_k}^+ + \theta_{xx_s} C_{x_s u_k}^+ + \theta_{xu} U_0$$

$$C_{x_s u_{k+1}}^- = C_{x_s u_k}^+$$

$$C_{xv_{k+1}}^- = \phi C_{xv_k}^+ + \theta_{xx_s} C_{x_s v_k}^+$$

$$C_{x_s v_{k+1}}^- = C_{x_s v_k}^+$$

The measurement residual covariance matrix is given by

$$J_{k+1} = H_{k+1} A_{k+1} + M_{k+1} B_{k+1} + G_{k+1} D_{k+1} + L_{k+1} E_{k+1} + R_{k+1}$$

where

$$A_{k+1} = P_{k+1}^- H_{k+1}^T + C_{xx}^- M_{k+1}^T + C_{xu}^- G_{k+1}^T + C_{xv}^- L_{k+1}^T$$

$$B_{k+1} = P_{s_{k+1}}^- M_{k+1}^T + C_{xx}^- H_{k+1}^T + C_{xs}^- G_{k+1}^T + C_{sv}^- L_{k+1}^T$$

$$D_{k+1} = C_{xu}^- H_{k+1}^T + C_{xs}^- M_{k+1}^T + U_o G_{k+1}^T$$

$$E_{k+1} = C_{xv}^- H_{k+1}^T + C_{sv}^- M_{k+1}^T + V_o L_{k+1}^T$$

The covariance matrix partitions immediately after processing a measurement are given by the following update equations:

$$P_{k+1}^+ = P_{k+1}^- - K_{k+1} A_{k+1}^T - A_{k+1} K_{k+1}^T + K_{k+1} J_{k+1} K_{k+1}^T$$

$$C_{xx}^+ = C_{xx}^- - K_{k+1} B_{k+1}^T - A_{k+1} S_{k+1}^T + K_{k+1} J_{k+1} S_{k+1}^T$$

$$P_{s_{k+1}}^+ = P_{s_{k+1}}^- - S_{k+1} B_{k+1}^T - B_{k+1} S_{k+1}^T + S_{k+1} J_{k+1} S_{k+1}^T$$

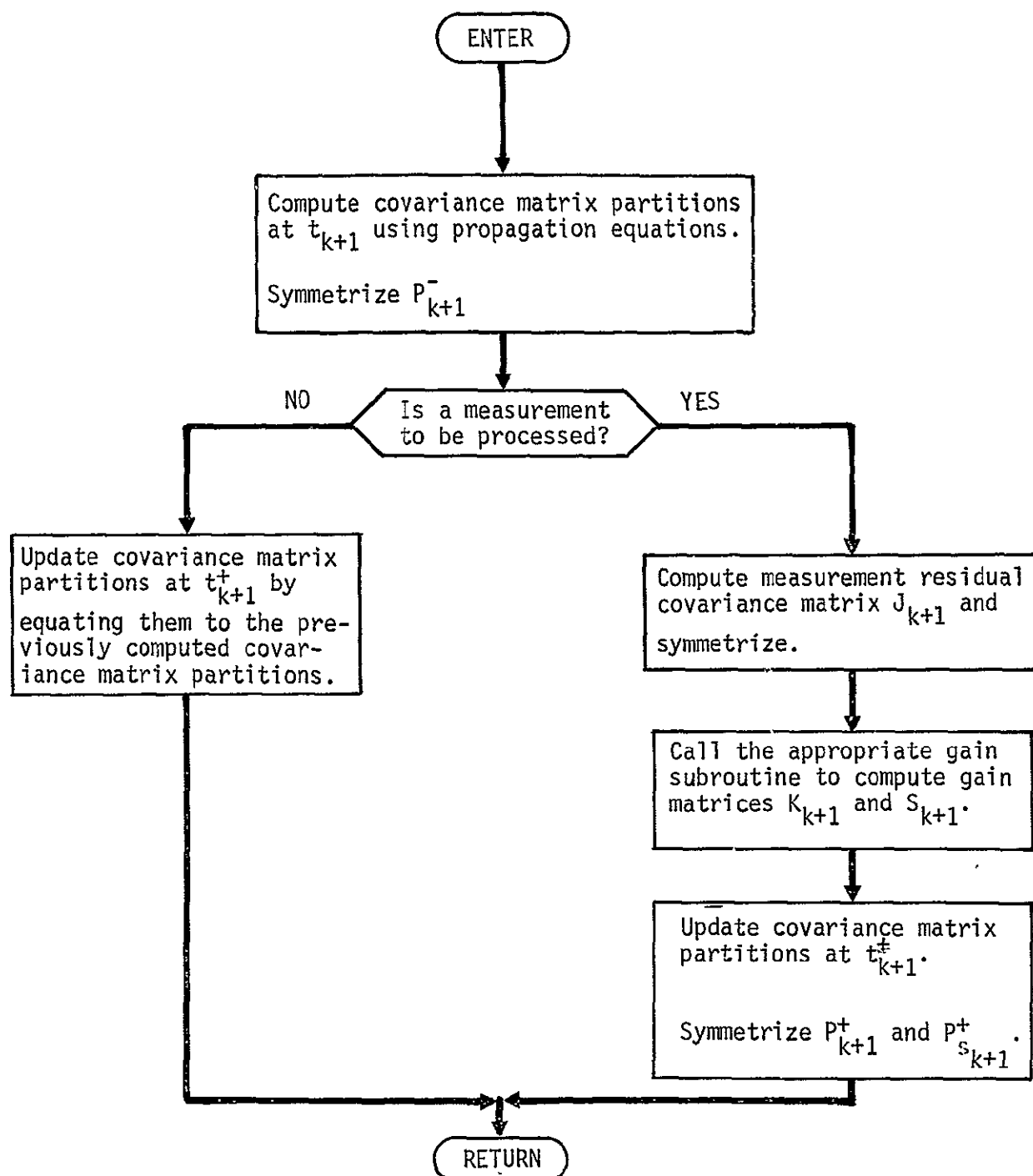
$$C_{xu}^+ = C_{xu}^- - K_{k+1} D_{k+1}^T$$

$$C_{xs}^+ = C_{xs}^- - S_{k+1} D_{k+1}^T$$

$$C_{xv}^+ = C_{xv}^- - K_{k+1} E_{k+1}^T$$

$$C_{sv}^+ = C_{sv}^- - S_{k+1} E_{k+1}^T$$

where gain matrices K_{k+1} and S_{k+1} are computed in the appropriate gain subroutine--GAIN1, if Kalman-Schmidt, GAIN2, if WLS.



SUBROUTINE NDTM

PURPOSE: TO COMPUTE THE UNAUGMENTED PORTION OF THE STATE TRANSITION MATRIX USING THE NUMERICAL DIFFERENCE TECHNIQUE.

CALLING SEQUENCE: CALL NDTM(RI,RF)

ARGUMENT: RF I POSITION AND VELOCITY OF THE VEHICLE AT THE END OF THE TIME INTERVAL

RI I POSITION AND VELOCITY OF THE VEHICLE AT THE BEGINNING OF THE TIME INTERVAL

SUBROUTINES SUPPORTED: PSIM

SUBROUTINES REQUIRED: NTM

LOCAL SYMBOLS: F1 TEMPORARY STORAGE FOR FACP
 F2 TEMPORARY STORAGE FOR FACV
 IPR INTERMEDIATE STORAGE FOR IPRINT
 RP POSITION AND VELOCITY OF VEHICLE AT INITIAL TIME
 SAVE TEMPORARY STORAGE FOR ACC
 T ALTERED POSITION AND VELOCITY OF VEHICLE AT INITIAL TIME
 U ALTERED POSITION AND VELOCITY OF VEHICLE AT FINAL TIME

COMMON COMPUTED/USED: ACC FACP FACV IPRINT

COMMON COMPUTED: PHI

COMMON USED: ACCND DELTM NDACC ONE ZERO

NDTM Analysis

The nonlinear equations of motion of the spacecraft can be written symbolically as

$$\dot{\vec{x}} = \vec{f}(\vec{x}, t) \quad (1)$$

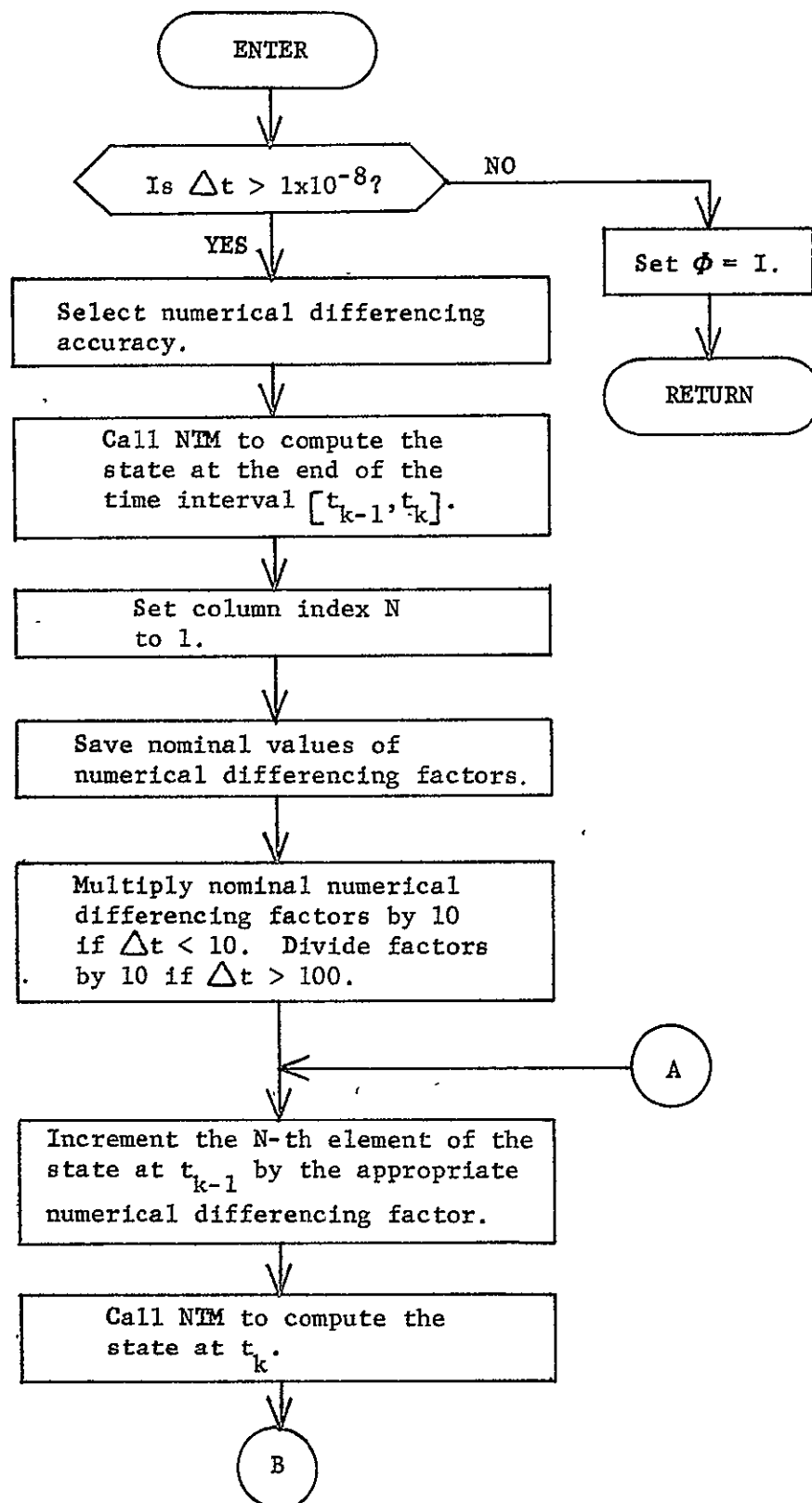
where \vec{x} is the spacecraft position/velocity state.

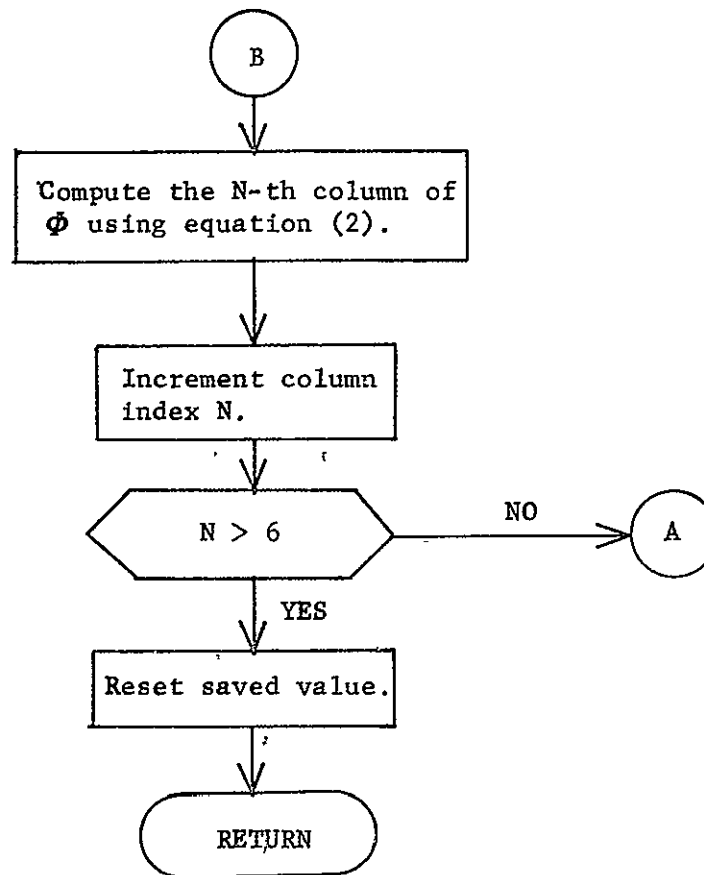
Suppose we wish to use numerical differencing to compute the state transition matrix $\Phi(t_k, t_{k-1})$. Let $\vec{\phi}(t_k, t_{k-1})$ represent the j -th column of $\Phi(t_k, t_{k-1})$. We assume we have available the nominal states $\vec{x}^*(t_{k-1})$ and $\vec{x}^*(t_k)$. To obtain $\vec{\phi}_j(t_k, t_{k-1})$ we increment the j -th element of $\vec{x}^*(t_{k-1})$ by the numerical differencing factor Δx_j and numerically integrate equation (1) over the time interval $[t_{k-1}, t_k]$ to obtain the new spacecraft state $\vec{x}_j(t_k)$. The j -subscript indicates $\vec{x}_j(t_k)$ was obtained by incrementing the j -th element of $\vec{x}^*(t_{k-1})$. Then

$$\vec{\phi}_j(t_k, t_{k-1}) = \frac{\vec{x}_j(t_k) - \vec{x}^*(t_k)}{\Delta x_j} \quad (2)$$

$$j = 1, 2, \dots, 6$$

NDTM Flow Chart





SUBROUTINE NEWPGE

PURPOSE: PRINTS APPROPRIATE HEADING AT THE TOP OF EACH PAGE WHEN
PRINTOUT OF TRAJECTORY INFORMATION IS DESIRED

CALLING SEQUENCE OALL NEWPGE

SUBROUTINES SUPPORTED INPUTZ PRINT SPACE

SUBROUTINES REQUIRED: NONE

COMMON COMPUTED/USED: IPG

COMMON COMPUTED: LINCT

COMMON USED: KL

PROGRAM NOMNAL

PURPOSE: TO CONTROL THE ENTIRE GENERATION OF A NOMINAL TRAJECTORY
 FROM INJECTION TARGETING THROUGH MIDCOURSE CORRECTIONS
 AND ORBIT INSERTION.

CALLING SEQUENCE: NONE (MAIN PROGRAM)

SUBROUTINES SUPPORTED: NONE (MAIN PROGRAM) .

SUBROUTINES REQUIRED: GIDANS PRELIM TRJTRY

COMMON COMPUTED: IPRE

COMMON USED: KWIT

NOMNAL Analysis

NOMNAL is the executive program controlling the entire generation of a nominal trajectory from injection targeting through midcourse corrections and orbit insertion.

NOMNAL begins by calling PRELIM for the preliminary work including initialization of variables, reading of the input data, and computation of zero iterate values of initial time, position, and velocity if required.

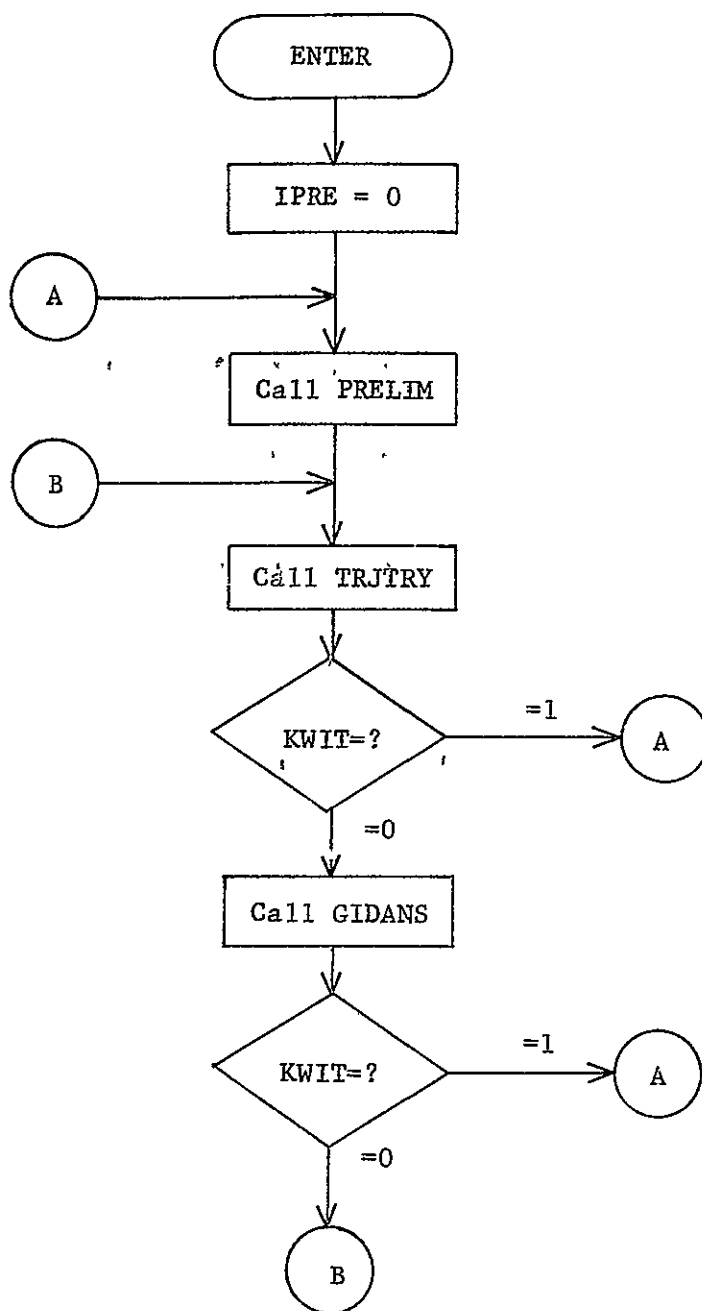
NOMNAL then calls TRJTRY. TRJTRY first determines the time of the next guidance event. It then integrates and records the nominal trajectory to that time. TRJTRY then returns control to NOMNAL.

NOMNAL next calls GIDANS. GIDANS processes the computation and execution of the current guidance event. NOMNAL then reenters its basic cycle by calling TRJTRY to propagate the corrected trajectory to its next guidance event.

Two flags are used by NOMNAL. The flag IPRE is initialized at zero in NOMNAL. During the processing of the first data case PRELIM sets it to unity. PRELIM uses IPRE to determine whether to preset constants to internally stored values or leave them at their previous values before reading the next data case.

The second flag KWIT determines whether the current case should be continued or terminated according to the flag value zero or unity respectively. Termination is indicated when a fatal error occurs during trajectory propagation or guidance event computation or when the desired end time is reached.

NOMNAL Flow Chart



SUBROUTINE NONINS

PURPOSE: TO DETERMINE THE TIME AND CORRECTION VECTOR FOR AN INSERTION FROM AN APPROACH HYPERBOLA INTO A SPECIFIED PLANE AND AS NEAR AS POSSIBLE TO A PRESCRIBED CLOSED ORBIT.

CALLING SEQUENCE: CALL NONINS(GM,X,Z,DA,DE,DWTP,DI,DN,TNEX,VEL,IEX)

ARGUMENT:	GM	I	GRAVITATIONAL CONSTANT
	X(3)	I	POSITION VECTOR AT DECISION
	Z(3)	I	VELOCITY VECTOR AT DECISION
	DA	I	DESIRED SEMIMAJOR AXIS
	DE	I	DESIRED ECCENTRICITY
	DWTP	I	DESIRED ARGUMENT OF PERIAPSIS
	DWTP	I	DESIRED ARGUMENT OF PERIAPSIS
	DI	I	DESIRED INCLINATION
	DN	I	DESIRED LONGITUDE OF ASCENDING NODE
	TNEX	O	TIME FROM DECISION TO EXECUTION (SECONDS)
	VEL(3)	O	INSERTION VELOCITY VECTOR
	IEX	O	EXECUTION CODE =0 EXECUTABLE SOLUTION DETERMINED =1 NO EXECUTABLE SOLUTION FOUND

SUBROUTINES SUPPORTED: INSEERS

SUBROUTINES REQUIRED: CAREL ELCAR

LOCAL SYMBOLS:	AH	HYPERBOLIC SEMIMAJOR AXIS
	ANG	TRUE ANOMALY OF HYPERBOLIC ASYMPTOTE
	ARC2	360.
	ARC	180.
	A	SEMIMAJOR AXIS OF MODIFIED ELLIPSE
	CEI	COSINE OF DI
	CEN	COSINE OF DN

CHI	COSINE OF HI
CHN	COSINE OF HN
CTAE	COSINE OF ETA
CTASY	COSINE OF TASY
CWTXE	COSINE OF WTXE
CWTXH	COSINE OF WTXH
DELV	VELOCITY CORRECTIONS OF CANDIDATE SOLUTION
DRA	DESIRED APOAPSIS RADIUS
DRP	DESIRED PERIAPSIS RADIUS
DTA	DUMMY VARIABLE FOR OUTPUT
DVM	MAGNITUDES OF CANDIDATE CORRECTIONS
DV	MAGNITUDES OF CANDIDATE CORRECTIONS
EH	HYPERBOLIC ECCENTRICITY
ERRMAX	SCALAR ERROR ASSIGNED TO IMPOSSIBLE SOLUTION
ERR	ARRAY OF SCALAR ERRORS OF SOLUTIONS
ETAX	TRUE ANOMALIES AT INTERSECTION POINTS ON ELLIPSE
E	ECCENTRICITY OF MODIFIED ELLIPSE
HI	HYPERBOLIC INCLINATION
HN	HYPERBOLIC LONGITUDE OF ASCENDING NODE
HRP	HYPERBOLIC PERIAPSIS RADIUS
HTAX	TRUE ANOMALIES AT INTERSECTION POINTS ON HYPERBOLA
HTA	CANDIDATE HYPERBOLIC TRUE ANOMALY AT INTERSECTION
MININ	INDEX OF OPTIMAL INTERSECTION POINT
MIN	INDEX OF OPTIMAL SOLUTION (POINT AND MOD)

NCPOS	FLAG INDICATING WHETHER ANGLE BETWEEN NODE AND INTERSECTION IS GREATER OR LESS THAN 180
NSOLS	NUMBER OF SOLUTIONS
NT1	INDEX OF FIRST SOLUTION
NT2	INDEX OF LAST SOLUTION
PH	HYPERBOLIC SEMILATUS RECTUM
PI	THE MATHEMATICAL CONSTANT PI
PP	UNIT VECTOR TOWARD PERIAPSIS
QQ	UNIT VECTOR IN ORBITAL PLANE NORMAL TO PP
RAD	DEGREE TO RADIAN FACTOR
RA	APOAPSIS RADIUS
RHYP	HYPERBOLIC CANDIDATE RADII TO INTERSECTION
RMAG	RADIUS TO INTERSECTION POINT
RM	RADIUS AT DECISION
RP	PERIAPSIS RADIUS
RX	RADIUS TO INTERSECTION POINT ON ELLIPSE
R1	RADIUS VECTOR TO HYPERBOLA AT INTERSECTION
R	RADIUS VECTOR TO ELLIPSE AT INTERSECTION
SEI	SINE OF DI
SEN	SINE OF DN
SGNZ	SIGN OF DECLINATION OF INTERSECTION POINT
SHI	SINE OF HI
SHN	SINE OF HN
STA	TRUE ANOMALY AT DECISION
TAE	ARRAY OF ELLIPTIC TRUE ANOMALIES
TASY	TRUE ANOMALY OF ASYMPTOTE
TAXE	TRUE ANOMALY AT INTERSECTION POINT ON

ELLIPSE

TAXH	TRUE ANOMALY AT INTERSECTION POINT ON HYPERBOLA
TA	TRUE ANOMALY
TDEC	TIME FROM PERIAPSIS AT DECISION
TEXC	TIME FROM PERIAPSIS AT EXECUTION ON HYPERBOLA
TEX	ARRAY OF CANDIDATE TIMES FROM DECISION TO EXECUTION
TTF	TIME FROM PERIAPSIS AT DECISION ON ELLIPSE
VM	SPEED AT DECISION
V1	VELOCITY VECTOR ON HYPERBOLA AT EXECUTION
V	VELOCITY VECTOR ON ELLIPSE AT EXECUTION
WE	ARGUMENT OF PERIAPSIS ON ELLIPSE
WH	ARGUMENT OF PERIAPSIS ON HYPERBOLA
WTXE	ANGLE BETWEEN ASCENDING NODE AND INTERSECTION POINT ON ELLIPSE
MTXH	ANGLE BETWEEN ASCENDING NODE AND INTERSECTION POINT ON HYPERBOLA
WN	UNIT NORMAL TO ORBITAL PLANE
W	ARGUMENT OF PERIAPSIS
XINT	X COMPONENT OF INTERSECTION POINT
YINT	Y COMPONENT OF INTERSECTION POINT
ZINT	Z COMPONENT OF INTERSECTION POINT

NONINS Analysis

NONINS determines the time and correction vector for an impulsive insertion from an approach hyperbola into a specified plane and as near as possible to a prescribed closed orbit. The approach hyperbola is specified by giving the planetocentric equatorial state \vec{r}, \vec{v} at the time of decision t_d . The final orbit is defined by giving its desired orbital elements $(a_E, e_E, i_E, \omega_E, \Omega_E)$ again in planetocentric equatorial coordinates.

Subroutine CAREL is first called to convert the hyperbolic state at decision \vec{r}, \vec{v} into Keplerian conic elements $(a_H, e_H, i_H, \omega_H, \Omega_H, t_{Hd})$ where t_{Hd} is the time from periapsis at decision (negative on the approach ray).

The points of intersection of the approach orbital plane and the desired orbital plane are then determined. The elements defining the two planes are therefore given by i_H, Ω_H and i_E, Ω_E . Let \hat{A} denote the unit vector toward the ascending node of an orbit and \hat{B} denote the in-plane normal to \hat{A} in the direction of motion. Then

$$\hat{A} = (\cos \Omega, \sin \Omega, 0) \quad (1)$$

$$\hat{B} = (-\sin \Omega \cos i, \cos \Omega \cos i, \sin i) \quad (2)$$

Hence the normal to the orbital plane \hat{C} is given by $\hat{C} = \hat{A} \times \hat{B}$ or

$$\hat{C} = (\sin \Omega \sin i, -\cos \Omega \sin i, \cos i) \quad (3)$$

The direction of the line of intersection of the two planes is therefore determined by $\vec{X} = \hat{C}_H \times \hat{C}_E$ or

$$\begin{aligned} \vec{X} = & (\cos i_H \sin i_E \cos \Omega_E - \sin i_H \cos i_E \cos \Omega_H, \\ & \cos i_H \sin i_E \sin \Omega_E - \sin i_H \cos i_E \sin \Omega_H, \\ & \sin i_H \sin i_E (\cos \Omega_H \sin \Omega_E - \sin \Omega_H \cos \Omega_E)) \end{aligned} \quad (4)$$

Then the unit vector along the line of intersection toward the northern hemisphere is given by

$$\hat{X} = \text{sgn } \vec{X}_3 \cdot \frac{\vec{X}}{|\vec{X}|} \quad (5)$$

Therefore the true anomaly f_{HX} along the hyperbola at the northern intersection point is given by

$$\cos (\omega_H + f_{HX}) = \hat{X} \cdot \hat{A}_H \quad (6)$$

The true anomaly on the hyperbola at the southern point is therefore $f_{HX} + 180^\circ$. Note that there exists a region of true anomalies lying between the incoming and outgoing asymptotes for which the hyperbola is not defined. Similar equations define the true anomaly on the ellipse at the two points of intersection. Note that this implies that the modified ellipse will have the same ω as the desired ellipse.

For the intersection true anomaly f_{HX} the radius magnitude on the hyperbola may be determined

$$r_I = \frac{a_H(1 - e_H^2)}{1 + e_H \cos f_H} \quad (7)$$

To permit an impulsive insertion, a_E and e_E must be modified to satisfy

$$r_I = \frac{a_E(1 - e_E^2)}{1 + e_E \cos f_E} \quad (8)$$

There are three candidate modifications examined to determine a "best"

- one:
- (1) Vary r_a while holding r_p constant
 - (2) Vary r_p while holding r_a constant
 - (3) Vary a while holding e constant

"Best" is defined below in terms of a weighted scalar function of the changes in r_a and r_p .

Rewriting (8) in terms of r_a and r_p (using $a = \frac{r_a + r_p}{2}$, $e = \frac{r_a - r_p}{r_a + r_p}$) yields the useful relation

$$r_a(1 + \cos f_E) + r_p(1 - \cos f_E) = \frac{2r_a r_p}{r_I} \quad (9)$$

Equation (9) may be solved for r_a as

$$r_a = \frac{r_I r_p (1 - \cos f_E)}{2r - r(1 + \cos f)} \quad (10)$$

This yields the r_a which defines the modified orbit holding r_p at its desired value. The semi-major axis and eccentricity are then computed from

$$a = \frac{r_a + r_p}{2}, \quad e = \frac{r_a - r_p}{r_a + r_p}.$$

Similarly (9) may be solved for r_p as

$$r_p = \frac{r_I r_a (1 + \cos f)}{2 r_a - r_I (1 - \cos f)} \quad (11)$$

This determines the modification is r_p required to achieve an intersecting ellipse having the desired r_a .

Finally (8) may be solved trivially for the a_E required to produce intersection for the desired eccentricity.

$$a_E = \frac{r_I (1 + e_E \cos f_E)}{(1 - e_E^2)} \quad (12)$$

An error is assigned to each of the candidate solutions as

$$E_i = W_i \left[|\Delta r_a| + |\Delta r_p| \right]$$

where Δr_a , Δr_p are the errors between the desired and modified values of r_a and r_p . The weighting factor W_i is assigned rather arbitrarily. Currently the weighting factor is $W_i = w_{1i} w_{2i}$ where

$$\begin{aligned} w_{1i} &= 1 && \text{if the true anomaly is on the incoming ray} \\ &2 && \text{if the true anomaly is on the outgoing ray} \\ w_{2i} &= 1 && \text{if option 1} \\ &2 && \text{if option 2} \\ &3 && \text{if option 3} \end{aligned}$$

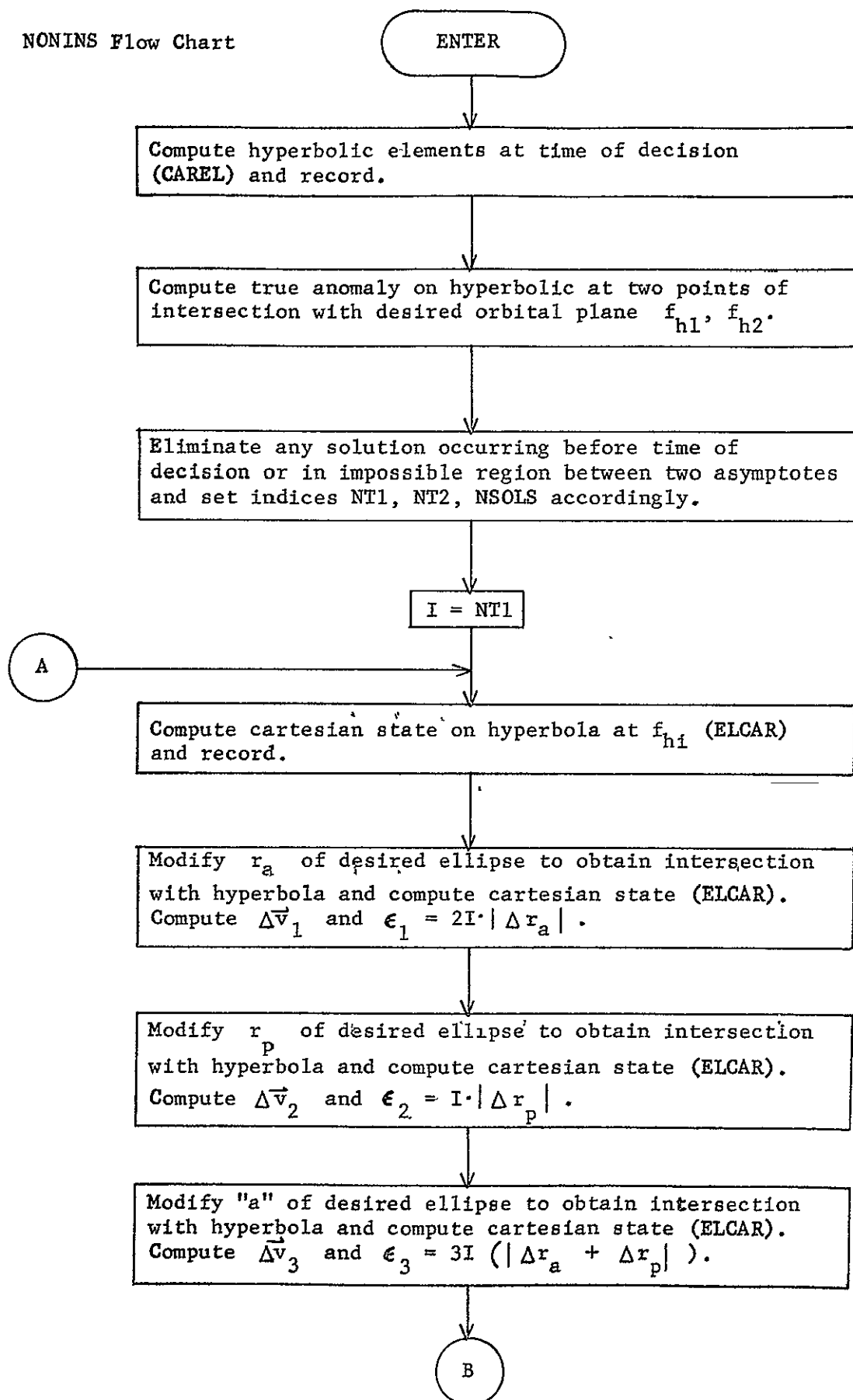
Thus a solution on the incoming asymptote is preferred over one on the outgoing asymptote and one subsequent trim is preferred over two subsequent trims.

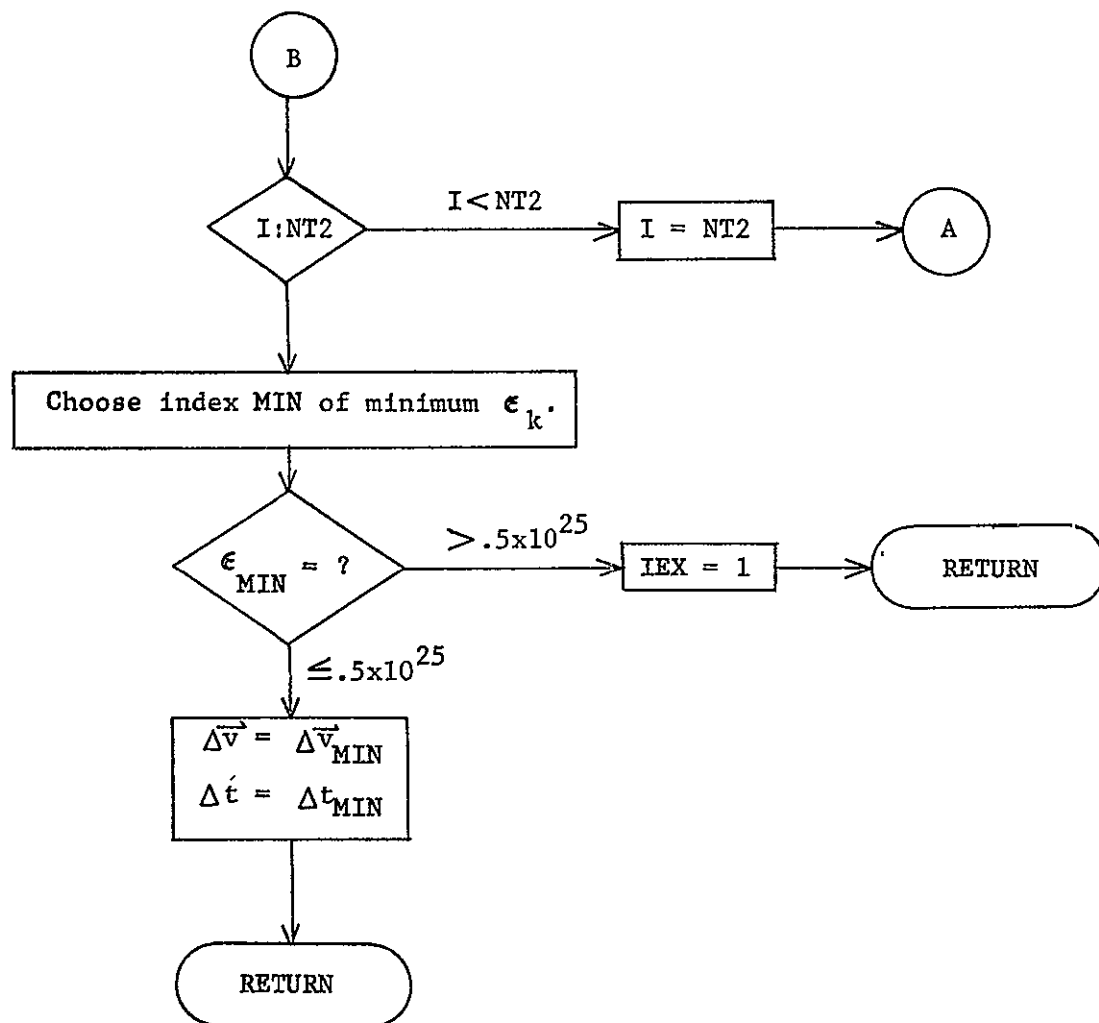
Having determined the elements of an intersecting orbit the insertion parameters are easily computed. The velocity on the hyperbola at the intersection point may be computed from ELCAR as \vec{v}_H . The velocity on the ellipse following the insertion is computed by calling ELCAR with the modified elliptical element to get \vec{v}_E . The impulsive $\Delta \vec{v}$ is then given by

$$\Delta \vec{v} = \vec{v}_E - \vec{v}_H$$

The time interval from the decision to the execution is given by the hyperbolic time from the initial point to the relevant intersection point.

NONINS Flow Chart





SUBROUTINE NONLIN

PURPOSE: TO CONTROL EXECUTION OF NON-LINEAR GUIDANCE EVENTS.

CALLING SEQUENCE: CALL NONLIN

SUBROUTINES SUPPORTED: GUISIM GUIDM

SUBROUTINES REQUIRED: CAREL ELCAR GIDANS

LOCAL SYMBOLS: AA ARGUMENT FOR SUBROUTINE CAREL
DI JULIAN DATE OF EVENT
EE ARGUMENT FOR SUBROUTINE CAREL
ISNPR SAVE INPR VALUE
ISPRNT SAVE IPRINT VALUE
KEY INTERMEDIATE VARIABLE IN SETTING UP TARGET
ARRAY
KICL2 SAVE ICL2 VALUE
KICL SAVE ICL VALUE
KISPH SAVE ISPH VALUE
KISP2 SAVE ISP2 VALUE
ODELT SAVE ORIGINAL DELTP VALUE
OSPH SAVE ORIGINAL SPHERE OF INFLUENCE OF
TARGET PLANET
PP ARGUMENT RETURNED FROM CAREL
QQ ARGUMENT RETURNED FROM CAREL
RMAG ARGUMENT FOR SUBROUTINE ELCAR
TAA TRUE ANOMALY
TFFP TIME OF FLIGHT FROM PERIAPSIS
TFP TIME FROM PERIAPSIS
TRTIME TRAJECTORY TIME OF THE GUIDANCE EVENT
VMAG ARGUMENT FOR SUBROUTINE ELCAR
WH ARGUMENT FOR SUBROUTINE CAREL

XXI ARGUMENT FOR SUBROUTINE CAREL
 XXN ARGUMENT FOR SUBROUTINE CAREL
 XYZTAA ZERO TRUE ANOMALY ARGUMENT FOR SUBROUTINE
 ELCAR

COMMON COMPUTED/USED:	DELTP	DELV	DSI	DT	IBADS
	ICL2	ICL	INPR	IPRINT	ISPH
	ISP2	KLP	KMXQ	KTAR	KTIM
	KTP	KWIT	MAT	MAXB	HDL
	NOIT	NTP	RSI	SPHERE	TAR
	TGT3	TIN	TMU	TOL	VSI
	XDC	XDELV	XRC	ZDAT	
COMMON COMPUTED:	ACKT	AC	BDR	BDT	DC
	DG	DVMAX	D1	ISTART	IZERO
	KTYP	KUR	LVLS	NLP	NOGYD
	NPAR	PERV	RC	RIN	SPHFAC
	TIMG	TRTH			
COMMON USED:	ACX	ALNGTH	DATEJ	DELTAV	IX
	JX	LKLP	LKTAR	LKTP	LLVLS
	LNPAR	PMASS	TM	T3	XAC
	XBDR	XBDT	XDSI	XDVMAX	XFAC
	XIN	XPERV	XRSI	XTAR	XTOL
	XVSI	ZERO			

NONLIN Analysis

NONLIN is the interface subroutine between the non-linear guidance subroutines of NOMNAL and subroutine GUIDM of ERRAN and subroutine GUISIM of SIMUL. NONLIN selects the necessary data from the ERRAN and SIMUL common blocks and stores into the common blocks of NOMNAL the information needed to compute and/or execute the $\Delta \vec{V}$ required in order to meet specified target conditions.

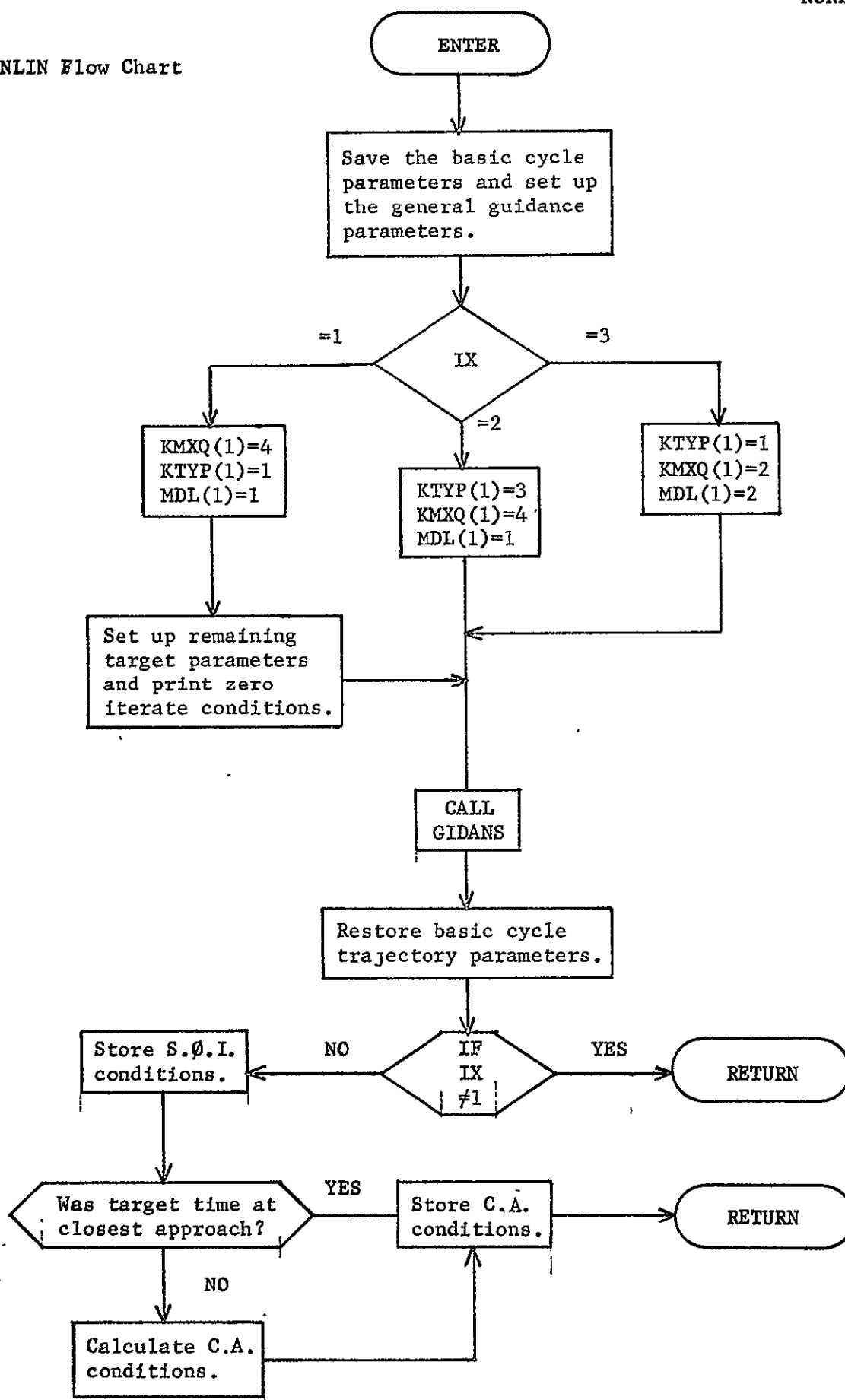
The most important task performed by NONLIN is the selection of the desired guidance scheme. The variable IX is tested and control is transformed according to the following:

- IX = 1, retargeting to specified target parameters
- = 2, orbit insertion to specified orbit
- = 3, $\Delta \vec{V}$ execution by a series of specified pulses

For each type of event, NONLIN then sets up values controlling the type of guidance event (KTYP), implementation code (KMXQ), and execution model code (MDL). For retargeting only, NONLIN stores the remaining values needed for $\Delta \vec{V}$ calculation and prints the zero iterate conditions.

NOMNAL calls GIDANS to perform the guidance event and restores parameters necessary for the basic cycles of ERRAN and SIMUL. For retargeting only, NOMNAL then stores the conditions at sphere of influence and closest approach of the target planet which were calculated by subroutine TARGET.

NONLIN Flow Chart



SUBROUTINE NTM

PURPOSE: CONTROL COMPUTATION OF NOMINAL TRAJECTORY IN THE ERROR ANALYSIS PROGRAM

CALLING SEQUENCE: CALL NTM(RI,RF,NTMC,ICODE)

ARGUMENT:

ICODE	I	INTERNAL CODE THAT DETERMINES WHICH TRAJECTORY IS BEING RUN AND WHAT INFORMATION IS DESIRED
NTMC	I	NOMINAL TRAJECTORY MODULE CODE THAT DETERMINES WHICH TYPE OF TRAJECTORY PROGRAM IS TO BE USED (NOTE ONLY THE VIRTUAL MASS TECHNIQUE IS SUPPLIED WITH THIS PROGRAM. HOWEVER, WITH LITTLE EFFORT ANY TRAJECTORY PROGRAM MAY BE ADDED AS AN EXTRA OPTION.)
RF	O	POSITION AND VELOCITY OF THE VEHICLE AT THE END OF THE TIME INTERVAL
RI	I	POSITION AND VELOCITY OF THE VEHICLE AT THE BEGINNING OF THE TIME INTERVAL

SUBROUTINES SUPPORTED: ERRANN MUND NDTM PLND PSIM
 SETEVN GUID VARADA PRED

SUBROUTINES REQUIRED: VMP

LOCAL SYMBOLS: D1 JULIAN DATE, EPOCH JAN.0, 1900, OF INITIAL TRAJECTORY TIME

RMP DISTANCE OF VEHICLE FROM TARGET PLANET AT SPHERE OF INFLUENCE OR CLOSEST APPROACH

VMM MAGNITUDE OF THE VELOCITY VECTOR

COMMON COMPUTED/USED:

BDRSI1	BDRSI2	BDRSI3	BDTSI1	BDTSI2
BDTSI3	BSI1	BSI2	BSI3	ICA1
ICA2	ICA3	ICL	ISOI1	ISOI2
ISOI3	ISPH	RCA1	RCA2	RCA3
RSOI1	RSOI2	RSOI3	TCA1	TCA2
TCA3	TSOI1	TSOI2	TSOI3	VSOI1
VSOI2	VSOI3			

COMMON USED:

ACC	BDR	BOT	B	DATEJ
DC	DELTM	DSI	IPROB	ISP2
ITR	NQE	RC	RSI	TRTN1

NTM Analysis

Subroutine NTM is used to generate the (most recent) targeted nominal trajectory in the error analysis mode. Subroutine NTM is equivalent to a subroutine NTMS from which all loops associated with ICODE = -3, -2, 2, 3 have been removed. For this reason no further analysis and no flow chart will be presented for subroutine NTM. Refer to subroutine NTMS.

SUBROUTINE NTMS

PURPOSE: CONTROL COMPUTATION OF TARGETED NOMINAL, MOST RECENT
NOMINAL, AND ACTUAL TRAJECTORIES IN THE SIMULATION
PROGRAM

CALLING SEQUENCE: CALL NTMS(RI,RF,NTMC,ICODE)

ARGUMENT:	ICODE	I	INTERNAL CODE THAT DETERMINES WHICH TRAJECTORY IS BEING RUN AND WHAT INFORMATION IS DESIRED
	NTMC	I	NOMINAL TRAJECTORY MODULE CODE THAT DETERMINES WHICH TYPE OF TRAJECTORY PROGRAM IS TO BE USED (NOTE ONLY THE VIRTUAL MASS TECHNIQUE IS SUPPLIED WITH THIS PROGRAM. HOWEVER, WITH LITTLE EFFORT ANY TRAJECTORY PROGRAM MAY BE ADDED AS AN EXTRA OPTION.)
	RF	O	POSITION AND VELOCITY OF THE VEHICLE AT THE END OF THE TIME INTERVAL
	RI	I	POSITION AND VELOCITY OF THE VEHICLE AT THE BEGINNING OF THE TIME INTERVAL

SUBROUTINES SUPPORTED: SIMULL MUND NDTM PLND PSIM
SETEVS GUISS VARSIM PRESIM

SUBROUTINES REQUIRED: VMP

LOCAL SYMBOLS:	ACCS	INTERMEDIATE STORAGE FOR ACCURACY
	D1	JULIAN DATE, EPOCH JAN., 1900, OF INITIAL TRAJECTORY TIME
	K1	INDEX FOR SEMIMAJOR AXIS ELEMENT
	K2	INDEX FOR ECCENTRICITY ELEMENT
	K3	INDEX FOR INCLINATION ELEMENT
	K4	INDEX FOR ASCENDING NODE ELEMENT
	K5	INDEX FOR PERIAPSIS ELEMENT
	K6	INDEX FOR MEAN ANOMALY ELEMENT
	NBODS	INTERMEDIATE STORAGE FOR NBOD
	NBS	INTERMEDIATE STORAGE FOR NB ARRAY
	RMP	DISTANCE OF VEHICLE FROM TARGET PLANET

SAVE10 INTERMEDIATE STORAGE
 SAVE1 INTERMEDIATE STORAGE
 SAVE2 INTERMEDIATE STORAGE
 SAVE3 INTERMEDIATE STORAGE
 SAVE4 INTERMEDIATE STORAGE
 SAVE5 INTERMEDIATE STORAGE
 SAVE6 INTERMEDIATE STORAGE
 SAVE7 INTERMEDIATE STORAGE
 SAVE8 INTERMEDIATE STORAGE
 VMH MAGNITUDE OF THE VELOCITY VECTOR

COMMON COMPUTED/USED: ACC BDRSI1 BDRSI2 BDRSI3 BDTSI1
 BDTSI2 BDTSI3 BSI1 BSI2 BSI3
 CN EMN ICA1 ICA2 ICA3
 ICL ISOI1 ISOI2 ISOI3 ISPH
 NBOD NB PMASS RCA1 RCA2
 RCA3 RSOI1 RSOI2 RSOI3 SHJR
 ST TCA1 TCA2 TCA3 TSOI1
 TSOI2 TSOI3 VSOI1 VSOI2 VSOI3

COMMON USED: ACC1 ALNGTH BDR BDT B
 DAB DATEJ DC DEB DELTH
 DIB DMAB DMUPB DMUSB DNBB
 DSI DWB IPROB ISP2 ITR
 NBOD1 NB1 NGE NQE NTP
 RC RSI TM TRTM1 VSI

NTMS Analysis

Subroutine NTMS is used to generate any of the three trajectories required in the simulation mode -- the (most recent) targeted nominal trajectory, the most recent nominal trajectory, and the actual trajectory.

The input variable ICODE is used to distinguish between these trajectories. It is unimportant to the virtual mass technique which trajectory is being computed. However, it is important to keep them separated so that the proper codes are set that check for approaching the sphere of influence of the target planet and reaching closest approach. It is also important to keep separate the conditions at which these occur for each trajectory. The following list describes ICODE completely.

ICODE = 3, NTMS will check to see if the sphere of influence and/or closest approach has been reached on the actual trajectory. If not, VMP will check for these conditions and on encountering either, NTMS places the conditions in special storage locations so they will be saved for future reference.

ICODE = 2, NTMS performs the same operations as described above for the most recent nominal trajectory.

ICODE = 1, NTMS again checks for sphere of influence and closest approach as above for the targeted nominal trajectory.

ICODE = 0, the only important information in this situation is the state vector at the end of the time interval. Therefore, NTMS does not check to see if closest approach or sphere of influence is encountered. This might occur in numerical differencing, for example.

ICODE = -1, it is important to know if sphere of influence or closest approach is reached on the targeted nominal trajectory. However, it is not desired that the information be stored for future use. This situation occurs in the guidance event.

ICODE = -2, the same comments may be made as if ICODE = -1, except this is on the most recent nominal trajectory.

ICODE = -3, again, this value of ICODE is treated the same as is ICODE = -1, for the actual trajectory.

Physical constants, planetary ephemerides, and other information relating to the dynamic model are the same for the targeted and most recent nominal trajectories. This is not true for the actual trajectory. There may be biases in the target planet ephemerides and the gravitational constants of the Sun and target planet. The numerical accuracy and the number of celestial bodies employed in the generation of the actual trajectory may also differ.

Ephemeris biases are specified as biases in orbital elements a , e , i , Ω , ω , and M .. However, within the program are stored the ephemeris constants of a , e , i , Ω , $\tilde{\omega}$, and M for the planets and a , e , i , Ω , $\tilde{\omega}$, and L for the moon, where

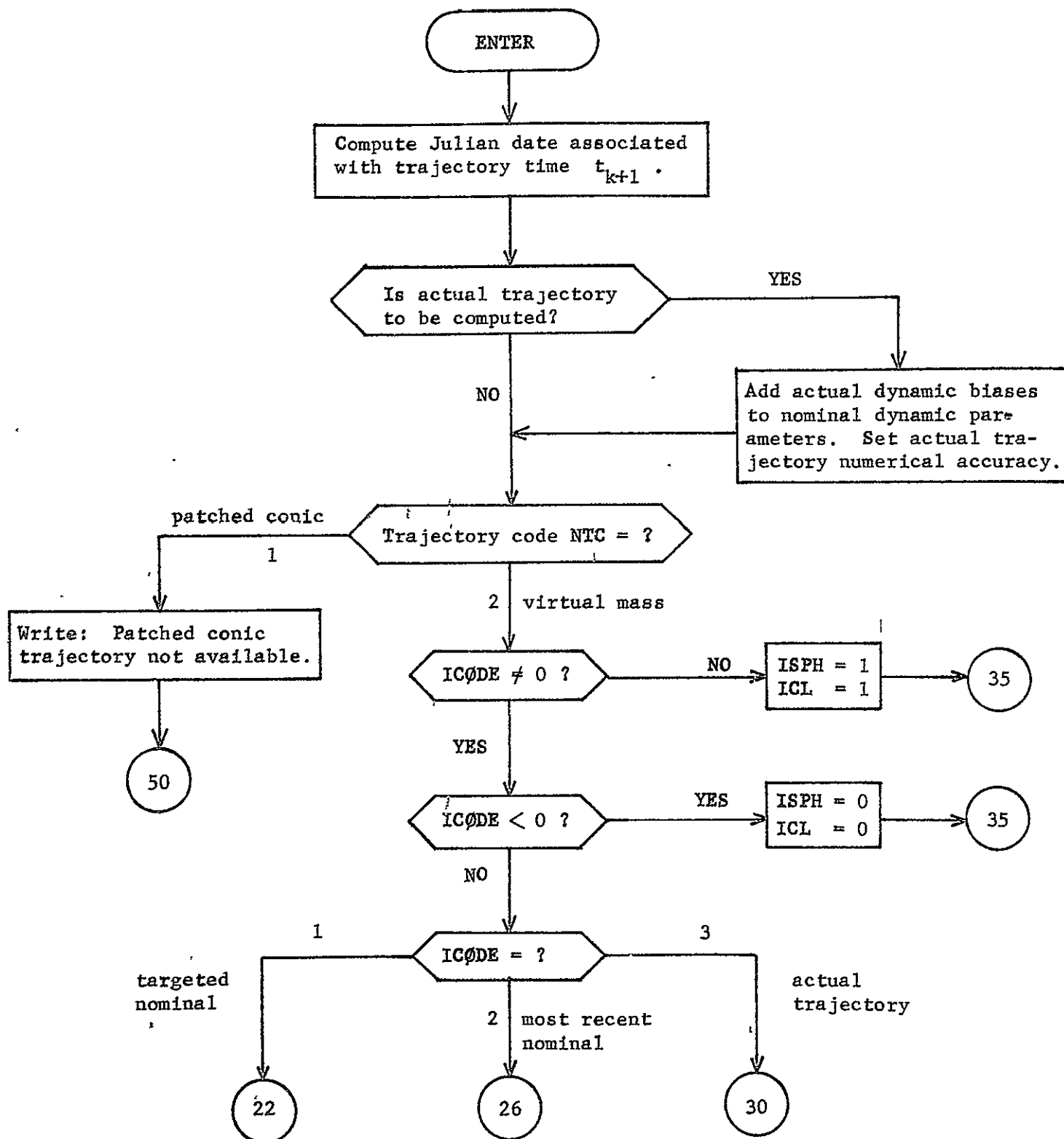
$$\tilde{\omega} = \omega + \Omega$$

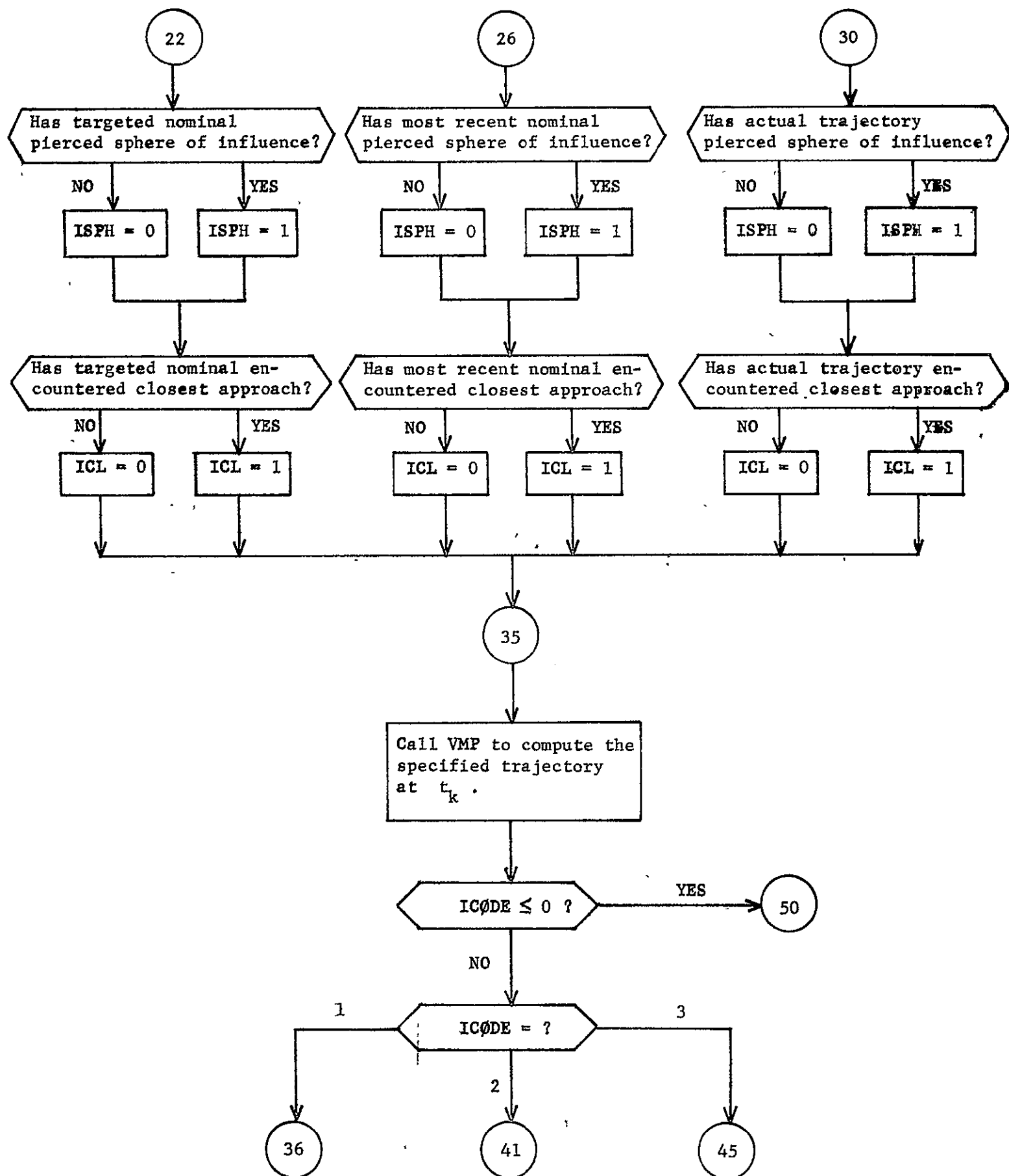
and

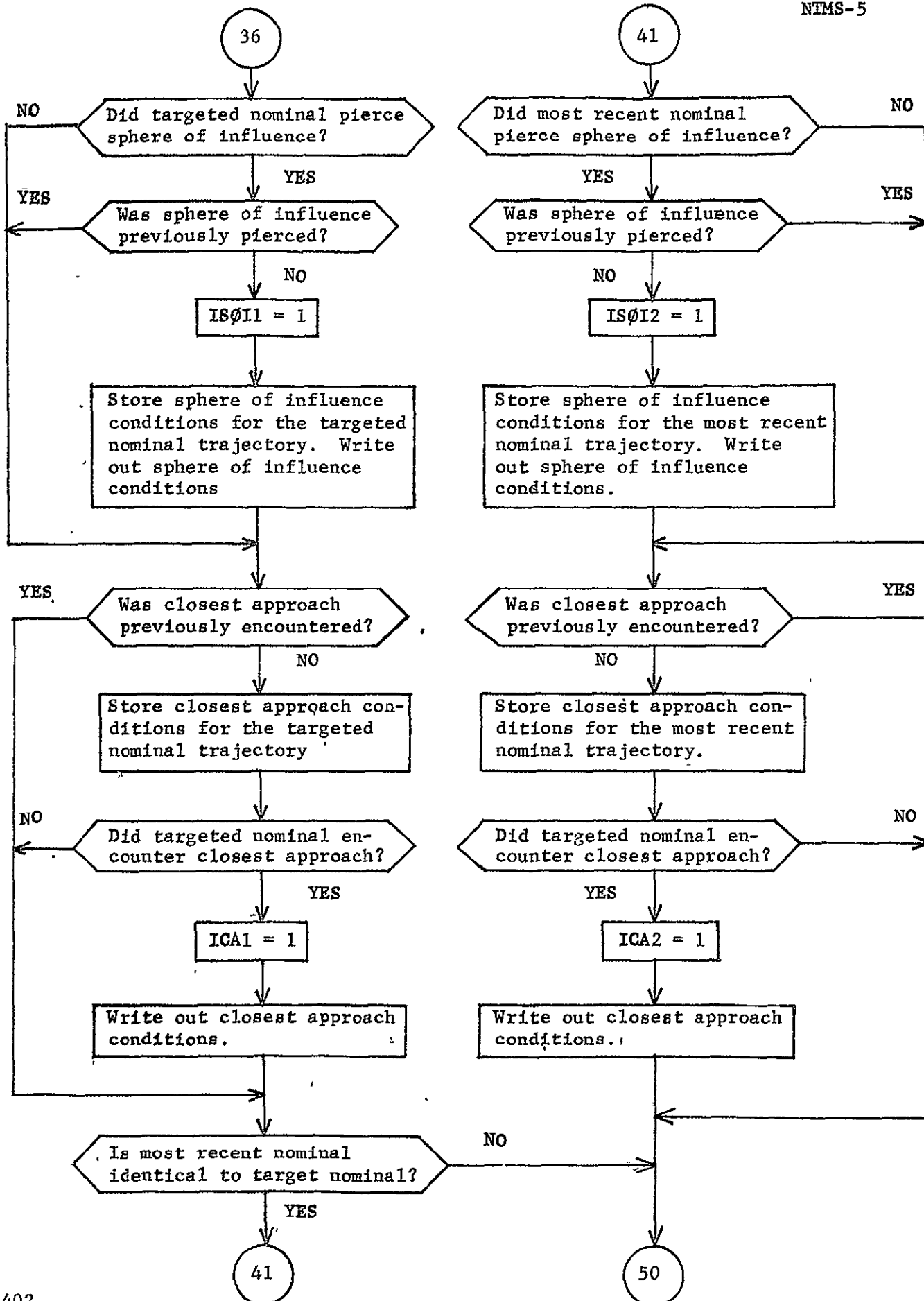
$$L = M + \omega + \Omega .$$

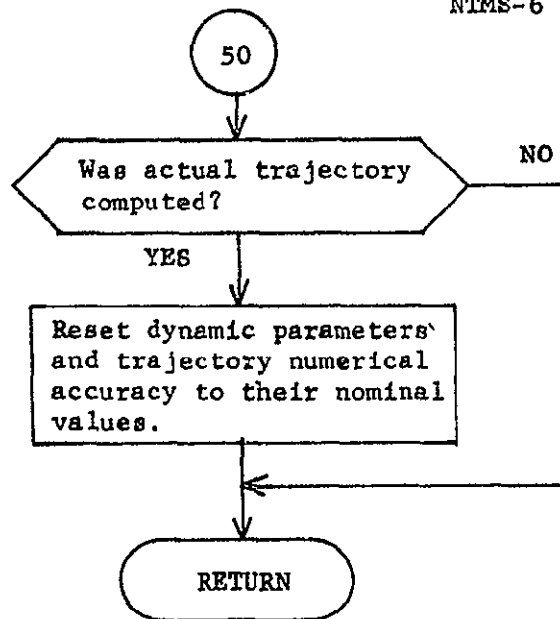
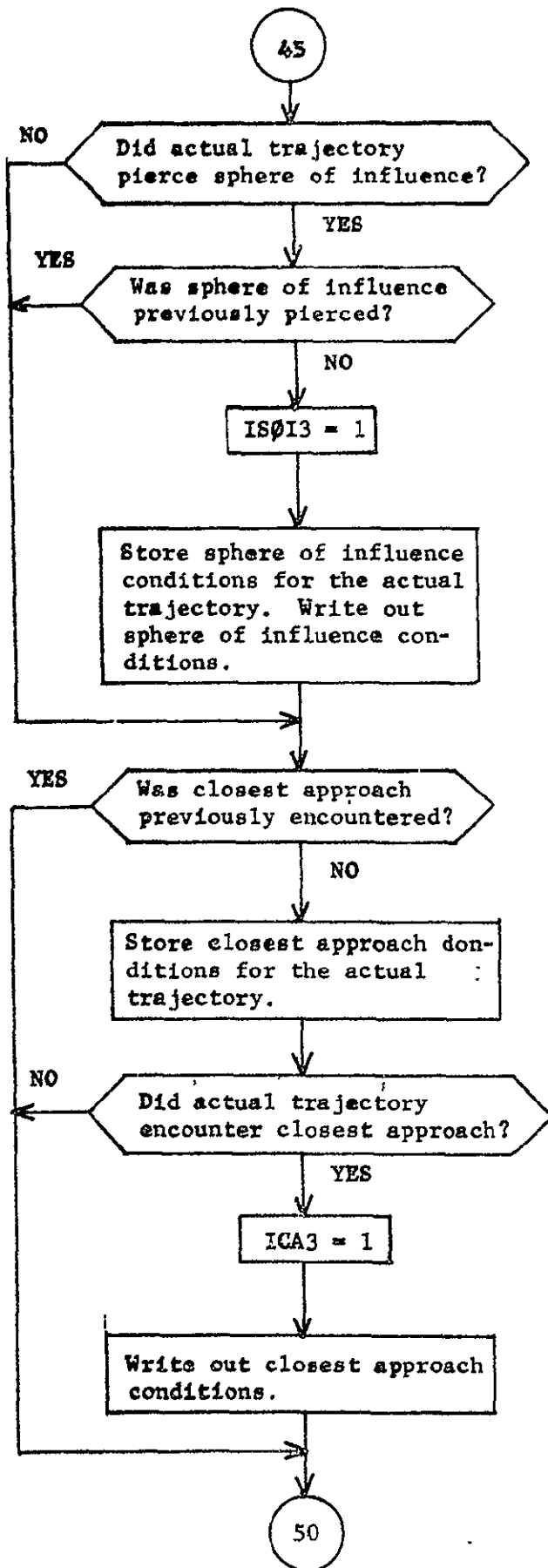
Incrementation of $\tilde{\omega}$ and L requires addition of biases in Ω , ω , and M as indicated by the above equations.

NTMS Flow Chart









SUBROUTINE NTRY

PURPOSE: TO COMPUTE ENTRY PARAMETERS, ENTRY COVARIANCE AND
COMMUNICATION ANGLE

CALLING SEQUENCE: CALL NTRY(TE,XPS,P,MODE)

ARGUMENTS: TE I ENTRY TIME (TRAJECTORY TIME)
XPS I PLANETOCENTRIC ECLIPTIC STATE AT PROBE
SPHERE
P I STATE COVARIANCE MATRIX
MODE I NOT FUNCTIONAL CURRENTLY

SUBROUTINES SUPPORTED: PROBE PROBES

SUBROUTINES REQUIRED: ORB EPHEM TIME SUBSOL MATPY

LOCAL SYMBOLS: A TRANSFORMATION FROM STEAP CARTESIAN STATE
VARIABLES TO LTR ENTRY PARAMETER STATE
VARIABLES
B INTERMEDIATE VARIABLE
H VEHICLE ALTITUDE
T TRANSFORMATION FROM PLANETOCENTRIC TO SUB-
SOLAR COORDINATES
V VEHICLE VELOCITY RELATIVE TO PLANET AT
PROBE SPHERE
AA INTERMEDIATE VARIABLE
ER UNIT VECTOR ALIGNED WITH SUBSOLAR RADIUS
VECTOR
EN VECTOR NORMAL TO ENTRY PLANE
HI ENTRY ORBIT ANGULAR MOMENTUM/UNIT MASS
MO MONTH
TZ JULIAN DATE CORRESPONDING TO TE
UP INTERMEDIATE VARIABLE
IHR HOUR
IYR YEAR

PPP	COMMUNICATION ANGLE IN DEGREES
SEC	SECONDS
SUM	INTERMEDIATE VARIABLE
XIS	INCLINATION OF ENTRY PLANE TO SUBSOLAR PLANE
XRE	POSITION OF EARTH AT TE
XVE	VELOCITY OF EARTH AT TE
XRP	POSITION OF TARGET PLANET AT TE
XVP	VELOCITY OF TARGET PLANET AT TE
COSP	COS(COMMUNICATION ANGLE)
FAC1	INTERMEDIATE VARIABLE
FAC2	INTERMEDIATE VARIABLE
FAC3	INTERMEDIATE VARIABLE
GAMA	FLIGHT PATH ANGLE
IDAY	DAY
IMIN	MINUTE
OMEGS	REFERENCE LONGITUDE OF ASCENDING NODE OF ENTRY PLANE RELATIVE TO SUBSOLAR COORDINATE SYSTEM
PHIPS	ANGLE BETWEEN ASCENDING NODE AND THE PHI REFERENCE LINE
PHIX	PHI REFERENCE (SET EQUAL TO ZERO)
PLTR	COVARIANCE MATRIX FOR LTR
PSAV	INTERMEDIATE ARRAY
RDOT	INTERMEDIATE VARIABLE
RSIS	SUBSOLAR POSITION COORDINATES OF SPACE CRAFT AT THE PROBE SPHERE
SUM1	INTERMEDIATE VARIABLE
SUM2	INTERMEDIATE VARIABLE

NTRY-C

SUM3 INTERMEDIATE VARIABLE

SUM4 INTERMEDIATE VARIABLE

SUM5 INTERMEDIATE VARIABLE

SUM6 INTERMEDIATE VARIABLE

VSIS SUBSOLAR VELOCITY COORDINATES OF SPACE
CRAFT AT THE PROBE SPHERE

BB INTERMEDIATE VARIABLE

COMMON COMPUTED/USED: NO XP

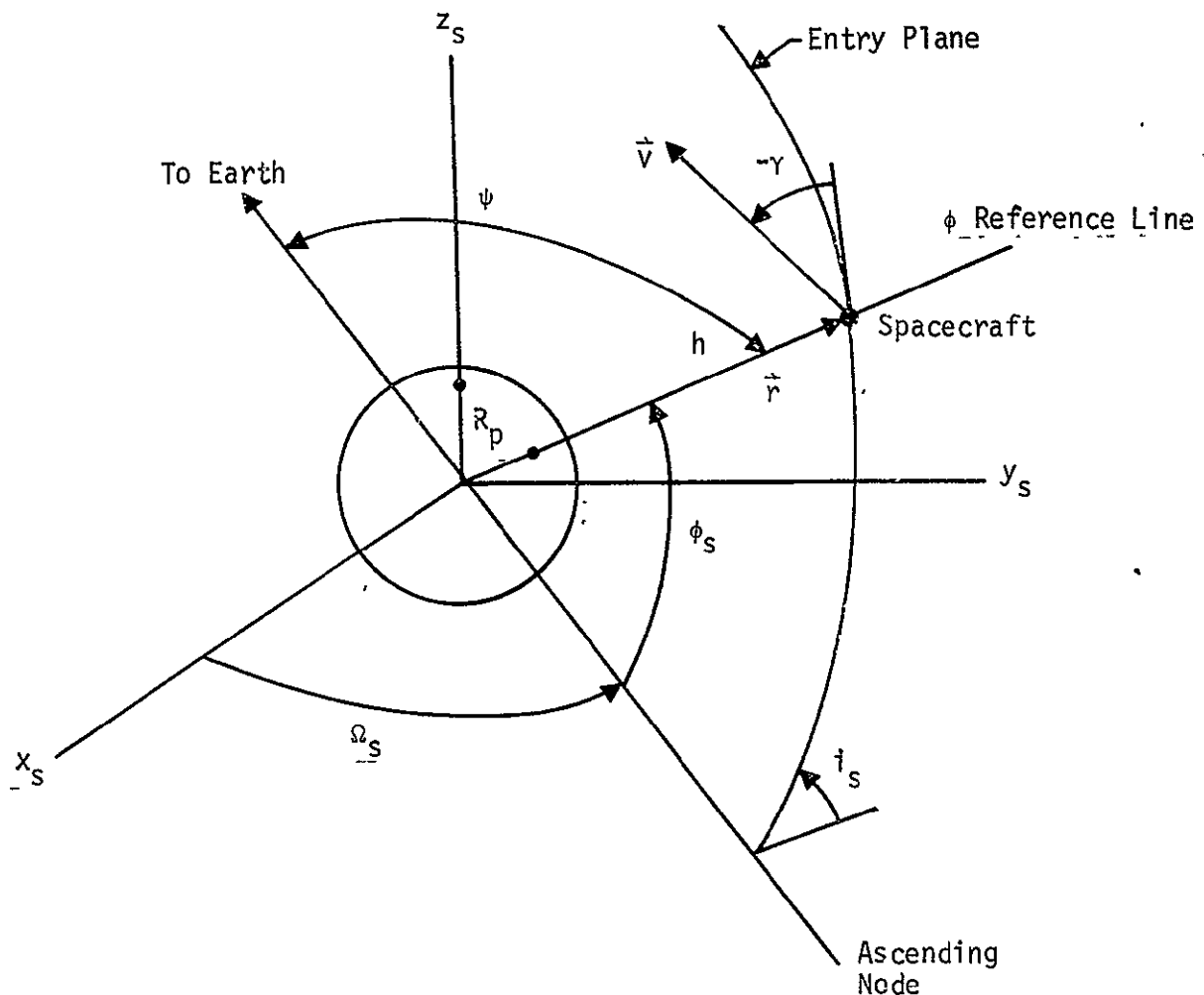
COMMON USED: DATEJ DSI NTP PI RAD
RPS TRTMB

NTRY Analysis

Subroutine NTRY transforms the heliocentric ecliptic spacecraft state and covariance matrix to entry parameter coordinates. This information is useful in defining initial data for the *Lander Trajectory Reconstruction (LTR)* program. Subroutine NTRY also computes the communication angle at entry.

The entry parameter state is defined by altitude h , velocity v relative to the planet, flightpath angle γ , longitude of the ascending node Ω_s , inclination i_s of the entry plane, and the angle ϕ_s between the ascending node and the ϕ reference line.

These latter angles are all defined relative to the subsolar orbital-plane coordinate system $x_s y_s z_s$, which is defined in subroutine SUBSOL. All entry parameters are shown in the following figure, as is the communication angle ψ .



The transformation of the heliocentric ecliptic spacecraft state to the entry parameter state requires first that the target planet heliocentric ecliptic state be subtracted to obtain the relative spacecraft position \vec{r} and velocity \vec{v} . The equations for transforming $\vec{r} = (r_x, r_y, r_z)$ and $\vec{v} = (v_x, v_y, v_z)$ to $(h, v, \gamma, \phi_s, i_s, \Omega_s)$ are summarized below.

Define $\vec{e}_r = \frac{\vec{r}}{r}$ and $\vec{e}_n = \frac{\vec{r} \times \vec{v}}{|\vec{r} \times \vec{v}|}$, where \vec{e}_r is a unit vector aligned with \vec{r} and \vec{e}_n is a unit vector normal to the entry plane. Let \vec{e}_z denote a unit vector aligned with the z_s -axis. The Cartesian subsolar orbital-plane components of these three unit vectors will be denoted as follows:

$$\vec{e}_r = (e_{r_x}, e_{r_y}, e_{r_z})$$

$$\vec{e}_n = (e_{n_x}, e_{n_y}, e_{n_z})$$

$$\vec{e}_z = (0, 0, 1).$$

Altitude h and velocity v are readily obtained:

$$h = |\vec{r}| - R_p \quad (1)$$

$$v = |\vec{v}| \quad (2)$$

where R_p is the target planet radius. Flightpath angle γ is computed from

$$\gamma = \sin^{-1} \left(\frac{\dot{r}}{v} \right) \quad (3)$$

where

$$\dot{r} = \vec{v} \cdot \vec{e}_r.$$

Longitude of the ascending node Ω_s is given by

$$\Omega_s = \tan^{-1} \left(\frac{e_{n_x}}{-e_{n_y}} \right), \quad (4)$$

while inclination i_s is obtained from

$$i_s = \cos^{-1} \left(e_{n_z} \right). \quad (5)$$

The angle ϕ_s is given by

$$\phi_s = \tan^{-1} \left(\frac{\sin \phi_s}{\cos \phi_s} \right) \quad (6)$$

where

$$\sin \phi_s = \frac{e_{r_z}}{\sin i_s}$$

and

$$\cos \phi_s = - \frac{((e_{n_y} e_{r_x} + e_{n_x} e_{r_y}))}{[e_{n_y}^2 + e_{n_x}^2]^{\frac{1}{2}}}.$$

If $i_s = 0$ or 180 degrees, the following equation for ϕ_s is used instead:

$$\phi_s = \tan^{-1} \left(\frac{e_{r_y}}{e_{r_z}} \right) - \Omega_s. \quad (7)$$

The desired entry parameter covariance matrix is defined by

$$P = E \left[\begin{matrix} \vec{x} & \vec{x}^T \\ \vec{x} & \vec{x}^T \end{matrix} \right] \quad (8)$$

where $\vec{x} = (\delta h, \delta v, \delta \gamma, \delta \phi_s)$. Given the covariance matrix

$$P' = E \left[\vec{x}' \vec{x}'^T \right] \quad (9)$$

where $\vec{x}' = (\delta r_x, \delta r_y, \delta r_z, \delta v_x, \delta v_y, \delta v_z)$, the desired covariance matrix can be obtained from

$$P = A P' A^T \quad (10)$$

where transformation matrix A is defined by

$$\vec{x} = A \vec{x}'. \quad (11)$$

The elements a_{ij} of the 4x6 matrix A are found by computing the differentials of equations (1), (2), (3), and (6). The results of this process are summarized as:

$$a_{11} = \frac{r_x}{r}, a_{12} = \frac{r_y}{r}, a_{13} = \frac{r_z}{r}, a_{14} = a_{15} = a_{16} = 0$$

$$a_{21} = a_{22} = a_{23} = 0, a_{24} = \frac{v_x}{v}, a_{25} = \frac{v_y}{v}, a_{26} = \frac{v_z}{v}$$

$$a_{31} = \frac{1}{h'} \left[v_x - \frac{r_x}{r^2} (\vec{r} \cdot \vec{v}) \right], a_{32} = \frac{1}{h'} \left[v_y - \frac{r_y}{r^2} (\vec{r} \cdot \vec{v}) \right],$$

$$a_{33} = \frac{1}{h'} \left[v_z - \frac{r_z}{r^2} (\vec{r} \cdot \vec{v}) \right], a_{34} = \frac{1}{h'} \left[r_x - \frac{v_x}{v^2} (\vec{r} \cdot \vec{v}) \right]$$

$$a_{35} = \frac{1}{h'} \left[r_y - \frac{v_y}{v^2} (\vec{r} \cdot \vec{v}) \right], a_{36} = \frac{1}{h'} \left[r_z - \frac{v_z}{v^2} (\vec{r} \cdot \vec{v}) \right]$$

$$a_{41} = -\frac{r_x \tan \phi_s}{r_z r^2}, a_{42} = -\frac{r_y \tan \phi_s}{r_z r^2}, a_{43} = \frac{\tan \phi_s}{r_z} \left(1 - \frac{r_z}{r^2} \right)$$

$$a_{44} = a_{45} = a_{46} = 0$$

where $h' = |\vec{r} \times \vec{v}|$ and is the orbit angular momentum/unit mass.

Communication angle ψ is computed from

$$\cos \psi = \frac{\vec{r} \cdot (\vec{r}_e - \vec{r}_p)}{|\vec{r}| \cdot |\vec{r}_e - \vec{r}_p|} \quad (12)$$

where

\vec{r} = spacecraft position relative to planet

\vec{r}_e = Earth position relative to sun

\vec{r}_p = planet position relative to sun.

SUBROUTINE ORB

PURPOSE: TO COMPUTE THE ORBITAL ELEMENTS -- INCLINATION,
LONGITUDE OF ASCENDING NODE, LONGITUDE OF PERIHELION,
ECCENTRICITY, AND LENGTH OF SEMIMAJOR AXIS -- FOR A
SPECIFIED PLANET AT A GIVEN TIME.

CALLING SEQUENCE: CALL ORB(IP,D)

ARGUMENT: D I JULIAN DATE, EPOCH 1900, OF THE TIME AT
WHICH THE ELEMENTS ARE TO BE CALCULATED

IP I CODE NUMBER OF PLANET
=1 SUN
=2 MERCURY
=3 VENUS
=4 EARTH
=5 MARS
=6 JUPITER
=7 SATURN
=8 URANUS
=10 PLUTO
=11 MOON

SUBROUTINES SUPPORTED: DATA DATAS PCTM PRINT3 PRINT4
PSIM TRAKM TRAKS TRAPAR VMP
GUIDM GUID GUISIM GUISS PRNTS3
HELIO LAUNCH LUNTAR MULCON MULTAR
PECEQ SAOCS SUBSOL PROBE PROBES
PRNTS4 TRAPAR

SUBROUTINES REQUIRED: NONE

LOCAL SYMBOLS: FN1 STATEMENT FUNCTION DEFINING A THIRD ORDER
POLYNOMIAL

FN2 STATEMENT FUNCTION DEFINING A FIRST ORDER
POLYNOMIAL

ITEMP INTERMEDIATE VARIABLE

PI2 TWICE THE MATHEMATICAL CONSTANT PI

COMMON COMPUTED/USED: ELMNT T

COMMON USED: CN EMN SMJR ST TWOPI

ORB Analysis

ORB determines the mean orbital elements for any gravitational body at a specified time.

The elements used are semi-major axis a , eccentricity e , inclination i , longitude of the ascending node Ω , and longitude of periapsis $\tilde{\omega}$. These elements are referenced to heliocentric ecliptic for the planets or geocentric ecliptic for the moon.

The mean elements are computed from time expansions as follows. Let α be any of the elements. Then the value of α at any time t is given by

$$\alpha(t) = \alpha_0 + \alpha_1 t + \alpha_2 t^2 + \alpha_3 t^3$$

where the constants α_k are stored by BLKDAT. These constants are stored into the arrays CN, ST, and EMN for inner planets, outer planets, and the moon respectively. The definitions of these arrays and the values stored are provided in the analysis of the previous subroutine BLKDAT. The element value as computed from the above equation is then returned in the ELMNT array according to the gravitational body code k as

ELMNT(8k-15) = i	$k = 1$ Sun	$= 1$ Saturn
ELMNT(8k-14) = Ω	2 Mercury	8 Uranus
ELMNT(8k-13) = $\tilde{\omega}$	3 Venus	9 Neptune
ELMNT(8k-12) = e	4 Earth	10 Pluto
ELMNT(8k-10) = a	5 Mars	11 Moon
ELMNT(8k-9) = ω	6 Jupiter	

SUBROUTINE PARTL

PURPOSE COMPUTE PARTIALS OF B DOT T AND B DOT R WITH RESPECT TO SPACECRAFT POSITION AND VELOCITY

CALLING SEQUENCE: CALL PARTL(R,V,B,BDT,BDR,PBT,PBR)

ARGUMENT: B 0 IMPACT PLANE PARAMETER
 BDR 0 B DOT R
 BDT 0 B DOT T
 PBR 0 PARTIAL OF B DOT R WITH RESPECT TO R AND V
 PBT 0 PARTIAL OF B DOT T WITH RESPECT TO R AND V
 R I POSITION OF VEHICLE RELATIVE TO PLANET
 V I VELOCITY OF VEHICLE RELATIVE TO PLANET

SUBROUTINES SUPPORTED: GUISS GUID

LOCAL SYMBOLS: H3 INTERMEDIATE VARIABLE
 RU INTERMEDIATE VARIABLE
 S MAGNITUDE OF VELOCITY
 U INTERMEDIATE VARIABLE
 U2 SQUARE OF U
 U2PV2 INTERMEDIATE VARIABLE
 UV INTERMEDIATE VARIABLE
 UV3 CUBE OF UV
 V2 SQUARE OF MAGNITUDE OF VELOCITY
 COMMON USED: ZERO

PARTL Analysis

PARTL is responsible for the computation of the partials of $B \cdot T$ and $B \cdot R$ with respect to the cartesian components of position and velocity.

Let the state of the spacecraft with respect to the target body at intersection with its sphere of influence be denoted

$$\vec{r} = [x, y, z]^T \quad r = \sqrt{x^2 + y^2 + z^2} \quad (1)$$

$$\vec{v} = [\dot{x}, \dot{y}, \dot{z}]^T \quad v = \sqrt{\dot{x}^2 + \dot{y}^2 + \dot{z}^2} \quad (2)$$

Introduce the approach asymptote \hat{S} and approximate it by the direction of \vec{v} .

$$\hat{S} = \frac{\vec{v}}{v} \quad (3)$$

The B-plane is the plane normal to \hat{S} containing the center of the target body. Any vector $\vec{\beta}$ within the B-plane must satisfy therefore

$$\hat{S} \cdot \vec{\beta} = 0 \quad (4)$$

The impact parameter vector \vec{B} is determined by the intersection of the B-plane and the incoming asymptote. The incoming asymptote is given parametrically by

$$\vec{\sigma} = \vec{r} + \vec{v} t \quad (5)$$

The time at which the asymptote intersects the B-plane may be determined by applying the B-plane condition (4)

$$\hat{S} \cdot \vec{r} + \hat{S} \cdot \vec{v} t = 0$$

$$t = - \frac{\vec{r} \cdot \vec{v}}{v^2} \quad (6)$$

Therefore the B-vector is given by

$$\vec{B} = \vec{r} - \frac{\vec{r} \cdot \vec{v}}{v^2} \vec{v}$$

$$\vec{B} = [x - \alpha \dot{x}, y - \alpha \dot{y}, z - \alpha \dot{z}]^T \quad (7)$$

where $\alpha = \frac{\vec{r} \cdot \vec{v}}{v^2} = \frac{x\dot{x} + y\dot{y} + z\dot{z}}{\dot{x}^2 + \dot{y}^2 + \dot{z}^2}$

Now assuming that the T axis is to lie in the x-y reference plane and the B-plane, it is defined as

$$\hat{T} = \frac{\hat{S} \times \hat{K}}{|\hat{S} \times \hat{K}|}$$

$$\hat{T} = \left[\frac{\dot{y}}{u}, -\frac{\dot{x}}{u}, 0 \right] \quad (8)$$

where $u^2 = \dot{x}^2 + \dot{y}^2$. The \hat{R} axis is defined by

$$\hat{R} = \hat{S} \times \hat{T}$$

$$\hat{R} = \frac{1}{uv} [\dot{x}\dot{z}, \dot{y}\dot{z}, -u^2]^T \quad (9)$$

Now combining (7), (8), (9) B·T and B·R may be computed in terms of the state components

$$B \cdot T = \frac{1}{u} (x\dot{y} - \dot{x}y)$$

$$B \cdot R = \frac{1}{uv} [(x\dot{x} + y\dot{y})\dot{z} - u^2 z] \quad (10)$$

where $u^2 = \dot{x}^2 + \dot{y}^2$, $v^2 = u^2 + \dot{z}^2$.

The partials may now be computed from differentiation of the above equations.

$$\begin{aligned} \frac{\partial B \cdot T}{\partial x} &= \frac{\dot{y}}{u} & \frac{\partial B \cdot R}{\partial x} &= \frac{\dot{x}\dot{z}}{uv} \\ \frac{\partial B \cdot T}{\partial y} &= -\frac{\dot{x}}{u} & \frac{\partial B \cdot R}{\partial y} &= \frac{\dot{y}\dot{z}}{uv} \\ \frac{\partial B \cdot T}{\partial z} &= 0 & \frac{\partial B \cdot R}{\partial z} &= -\frac{u}{v} \\ \frac{\partial B \cdot T}{\partial \dot{x}} &= -\frac{\dot{y}}{u^3} (x\dot{x} + y\dot{y}) & \frac{\partial B \cdot R}{\partial \dot{x}} &= \frac{\dot{z}}{u^3 v^3} \left[u^2 (v^2 x - \dot{x}\dot{z}) - \dot{x}(u^2 + v^2)(x\dot{x} + y\dot{y}) \right] \\ \frac{\partial B \cdot T}{\partial \dot{y}} &= \frac{\dot{x}}{u^3} (x\dot{x} + y\dot{y}) & \frac{\partial B \cdot R}{\partial \dot{y}} &= \frac{\dot{z}}{u^3 v^3} \left[u^2 (v^2 y - \dot{y}\dot{z}) - \dot{y}(u^2 + v^2)(x\dot{x} + y\dot{y}) \right] \\ \frac{\partial B \cdot T}{\partial \dot{z}} &= 0 & \frac{\partial B \cdot R}{\partial \dot{z}} &= \frac{u}{v^3} (x\dot{x} + y\dot{y} + z\dot{z}) \end{aligned}$$

SUBROUTINE PCTM

PURPOSE: CONTROL COMPUTATION OF STATE TRANSITION MATRIX USING THE
ANALYTICAL PATCHED CONIC TECHNIQUE

CALLING SEQUENCE: CALL PCTM(RI)

ARGUMENT: RI I POSITION AND VELOCITY OF THE VEHICLE AT THE
BEGINNING OF THE TIME INTERVAL

SUBROUTINES SUPPORTED: PSIM

SUBROUTINES REQUIRED: CONC2 EPHEM' ORB

LOCAL SYMBOLS: D JULIAN DATE, EPOCH JAN.0, 1900, OF INITIAL
TIME

DELT LENGTH OF TIME INCREMENT IN PROPER UNITS

DUM TEMPORARY STORAGE FOR STATE TRANSITION
MATRIX

GMS GRAVITATIONAL CONSTANT OF GOVERNING BODY

IP CODE OF PLANET

RM DISTANCE FROM SPECIFIED PLANET

RS POSITION OF VEHICLE RELATIVE TO SPECIFIED
PLANET

VS VELOCITY OF VEHICLE RELATIVE TO SPECIFIED
PLANET

COMMON COMPUTED/USED: XP

COMMON COMPUTED: NO PHI

COMMON USED: ALNGTH DATEJ DELTM F IBARY
NBOD NB PMASS SPHERE TM
TRTM1

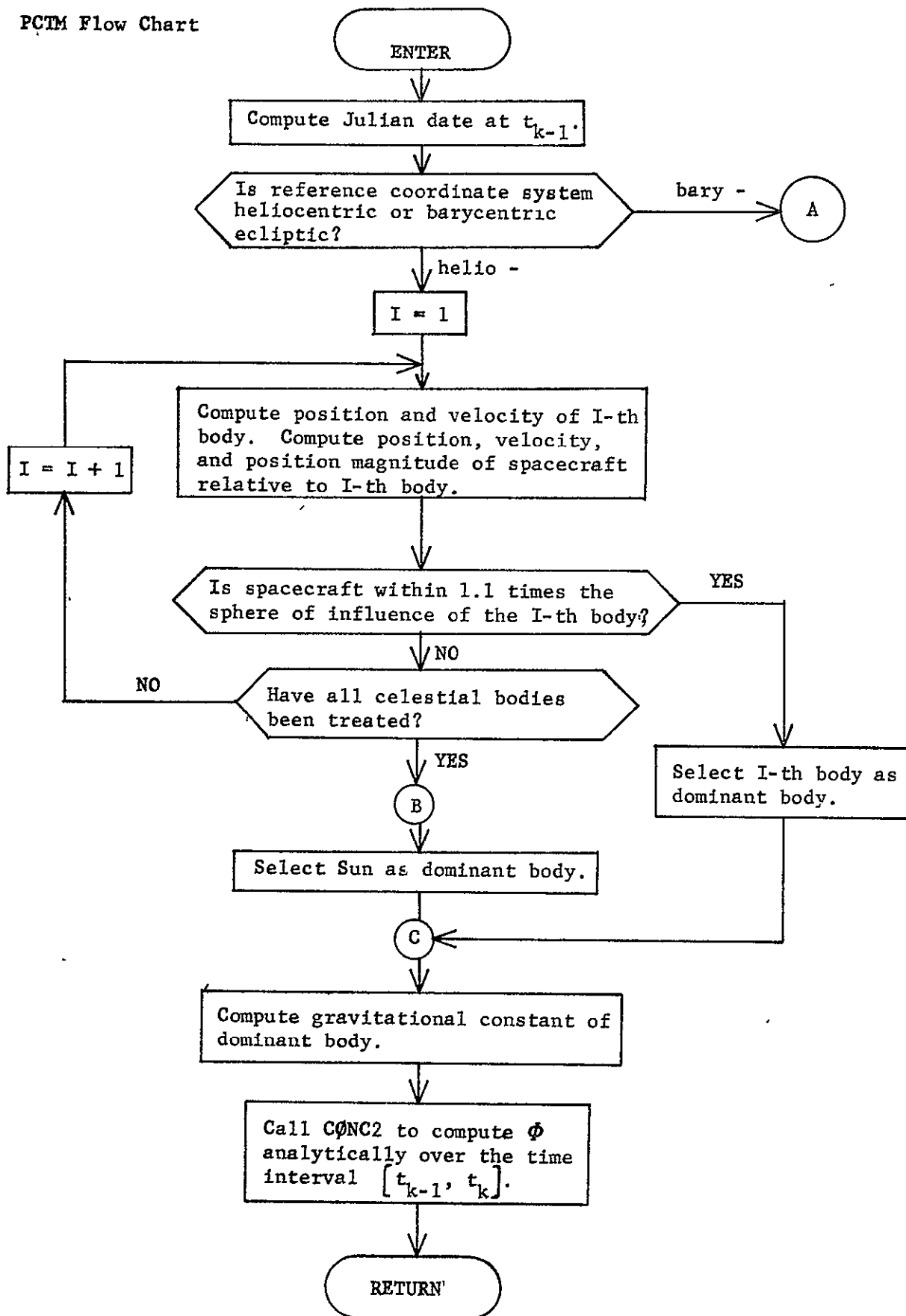
PCTM Analysis

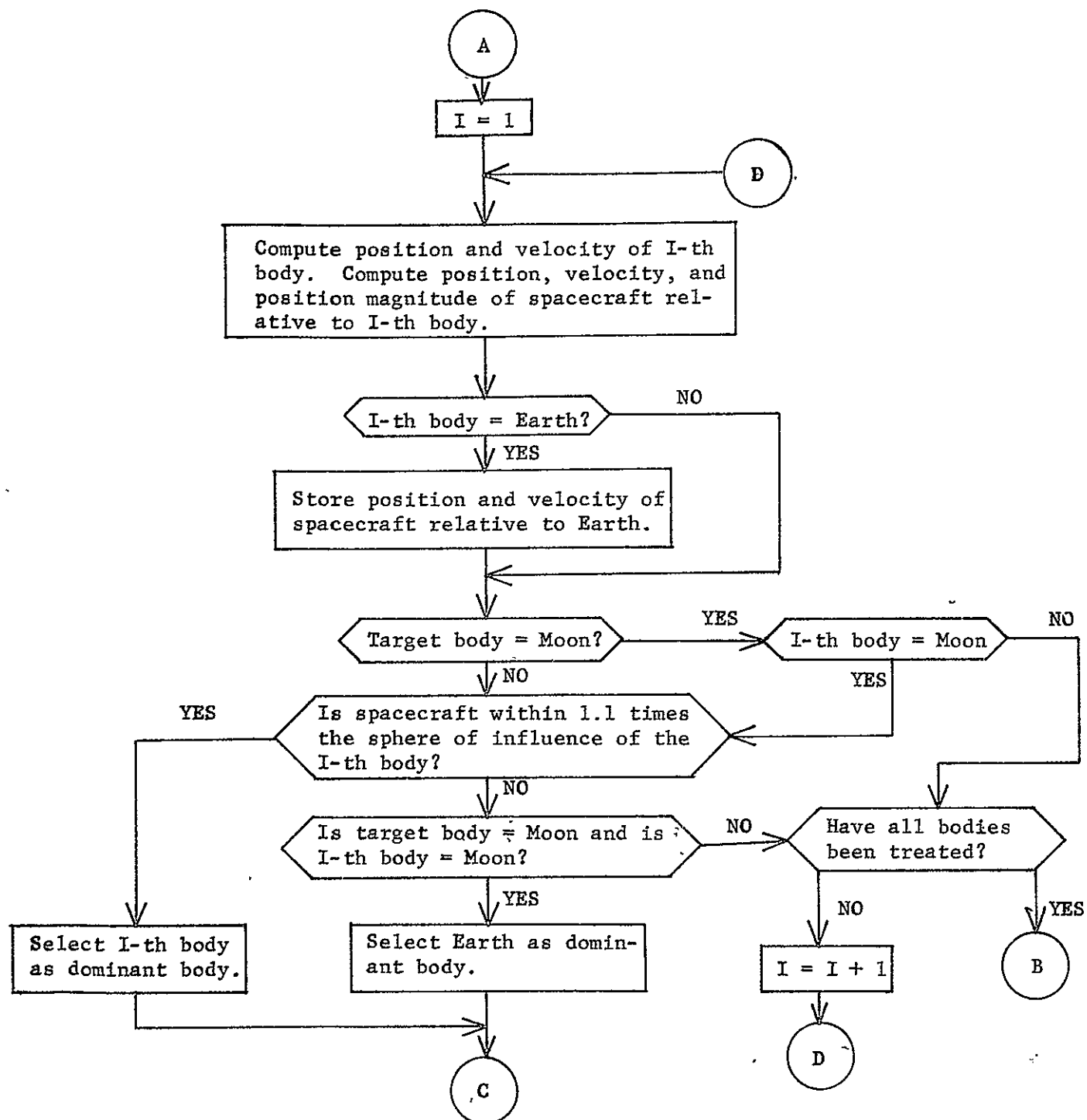
Subroutine PCTM does not actually compute the state transition matrix $\Phi(t_k, t_{k-1})$ itself; this is accomplished by calling CØNC2 from within PCTM. The primary function of PCTM is to determine the dominant body at time t_{k-1} to be used in the computation of $\Phi(t_k, t_{k-1})$ by means of the analytical patched conic technique.

On interplanetary trajectories we compute the distance separating the spacecraft from each of the celestial bodies included in the analysis. If the distance between the spacecraft and the i -th body is less than or equal to 1.1 times the sphere of influence of the i -th body, the i -th body is selected as the dominant body. Otherwise, the Sun is selected as the dominant body.

On lunar trajectories we compute the distance separating the spacecraft from the Moon. If this distance is less than or equal to 1.1 times the sphere of influence of the Moon, the moon is selected as the dominant body. If not, the Earth is selected as the dominant body.

PCTM Flow Chart





SUBROUTINE PECEQ

PURPOSE: TO COMPUTE THE MATRIX DEFINING THE TRANSFORMATION FROM
PLANET CENTERED ECLIPTIC COORDINATES TO PLANET CENTERED
EQUATORIAL COORDINATES AS A FUNCTION OF THE PARTICULAR
PLANET AND TIME.

CALLING SEQUENCE: CALL PECEQ(NP,D,ECEQ)

ARGUMENT NP I CODE OF PLANET
D I JULIAN DATE, EPOCH 1900, OF REFERENCE TIME
ECEQ(3,3) O COORDINATE TRANSFORMATION MATRIX FROM
PLANETOCENTRIC ECLIPTIC TO PLANETOCENTRIC
EQUATORIAL COORDINATES

SUBROUTINES SUPPORTED: TARGET HELIO LAUNCH LUNTAR MULTAR
INSERS TRAPAR VMP DATAS GUISIM
DATA GUIDM EXCUTE SOIPS TPRTRG

SUBROUTINES REQUIRED: EULMX ORB

LOCAL SYMBOLS: AGCAC COORDINATE TRANSFORMATION MATRIX FROM
ORBITAL PLANE TO EQUATORIAL COORDINATES
FOR MOON
AHGCG COORDINATE TRANSFORMATION MATRIX FROM
GEOCENTRIC ECLIPTIC TO GEOCENTRIC
EQUATORIAL COORDINATES FOR EARTH - FROM
ECLIPTIC TO ORBITAL PLANE COORDINATES FOR
MOON
CSDECL COSINE OF DECL
CSEOBL COSINE OF EOBL
CSINM COSINE OF INM
CSNDM COSINE OF NODEM
CSRASC COSINE OF RASC
DECL DECLINATION OF TARGET PLANET POLE
DGTR CONVERSION FACTOR FROM DEGREES TO RADIANS
ED JULIAN DATE , EPOCH 4713 B.C.
EOBL OBLIQUITY OF ECLIPTIC
INM INDEX

NODEM	INDEX
NORM	UNIT VECTOR NORMAL TO TARGET PLANET ORBITAL PLANE
PBAR	CROSS PRODUCT OF POLE AND NORM
PMAG	MAGNITUDE OF PBAR
POLE	UNIT VECTOR ALIGNED WITH TARGET PLANET POLAR AXIS
POLMAG	MAGNITUDE OF POLE
QBARP	CROSS PRODUCT OF POLE AND PBAR
QMAG	MAGNITUDE OF QBARP
RASC	RIGHT ASCENSION OF TARGET PLANET POLE
SNDECL	SINE OF DECL
SNEOBL	SINE OF EOBL
SNINM	SINE OF INCLINATION INM
SNNDM	SINE OF NODE NDM
SNRASC	SINE OF RASC
TPRIM	BESSELIAN DATE
XI	INTERMEDIATE VALUE
XIQ	INTERMEDIATE VALUE
XL	INTERMEDIATE VALUE
XLQ	INTERMEDIATE VALUE

COMMON USED:

EMN ONE T ZERO

PECEQ Analysis

Subroutine PECEQ computes the coordinate transformation maxtrix A from planetocentric ecliptic to planetocentric equatorial coordinates for an arbitrary planet.

The derivation of A for a planet other than the earth or moon will be summarized. Matrix A is defined by

$$A = \begin{bmatrix} \hat{X} & \hat{Y} & \hat{Z} \end{bmatrix}^T \quad (1)$$

where \hat{X} , \hat{Y} , and \hat{Z} are unit vectors aligned with the planetocentric equatorial coordinate axes and referenced to the planetocentric ecliptic coordinate system. Unit vector \hat{Z} is aligned with the planet pole. Unit vector \hat{X} lies along the intersection of the of the planet equatorial and orbital planes and points at the planet vernal equinox. Unit vector \hat{Y} completes the orthogonal triad and is given by

$$\hat{Y} = \hat{Z} \times \hat{X}. \quad (2)$$

It remains to obtain expressions for \hat{X} and \hat{Z} . Let \hat{N} denote the unit vector normal to the planet orbital plane, and let \hat{P} denote the unit vector aligned with the planet pole. Then

$$\hat{Z} = \hat{P} \quad (3)$$

and

$$\hat{X} = \frac{\hat{P} \times \hat{N}}{|\hat{P} \times \hat{N}|}. \quad (4)$$

The unit vector \hat{N} , referred to the ecliptic coordinate system, is given by

$$\hat{N} = \begin{bmatrix} \sin i \sin \Omega \\ -\sin i \cos \Omega \\ \cos i \end{bmatrix} \quad (5)$$

where i and Ω are the inclination and longitude of the ascending node, respectively, of the planet orbital plane. The unit vector \hat{P} , referred to the ecliptic system is given by

$$P = \begin{bmatrix} \cos \alpha \cos \delta \\ \cos \epsilon \sin \alpha \cos \delta + \sin \epsilon \sin \delta \\ -\sin \epsilon \sin \alpha \cos \delta + \cos \epsilon \sin \delta \end{bmatrix} \quad (6)$$

where α and δ are the right ascension and declination, respectively, of the planet pole relative to the geocentric equatorial coordinate system, and ϵ is the obliquity of the ecliptic. Expressions for α and δ for each planet were obtained from JPL TR 32-1306, *Constants and Related Information for Astrodynamic Calculations*, 1968, by Melbourne, et al.

For the earth and the moon, the transformation matrix A is written as the produce of two transformation matrices

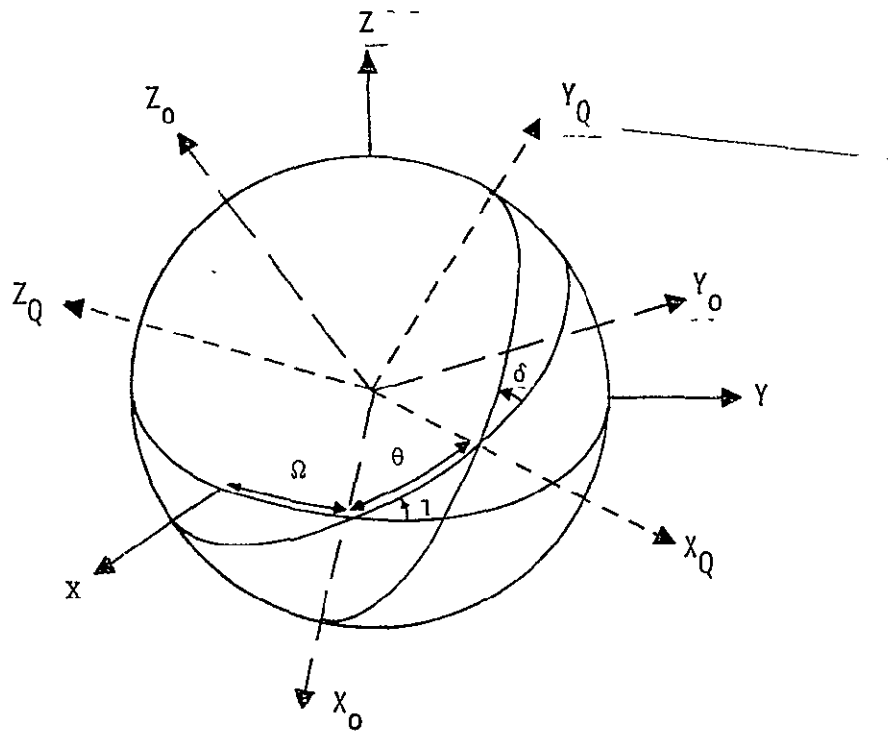
$$A = A_2 A_1. \quad (7)$$

For the earth A_2 is the identity matrix and A_1 is given by

$$A_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \epsilon & -\sin \epsilon \\ 0 & \sin \epsilon & \cos \epsilon \end{bmatrix}. \quad (8)$$

The following figure defines the transformations A_1 and A_2 , using the definitions given.

XYZ	Ecliptic coordinate axes
$X_O Y_O Z_O$	Orbital plane coordinate axes
$X_Q Y_Q Z_Q$	Moon's equatorial coordinate axes
i	Inclination of moon's orbital plane to ecliptic plane
Ω	Right ascension of moon's orbital plane to ecliptic plane
δ	Inclination of moon's equatorial to orbital plane
θ	Right ascension of moon's equatorial to orbital plane



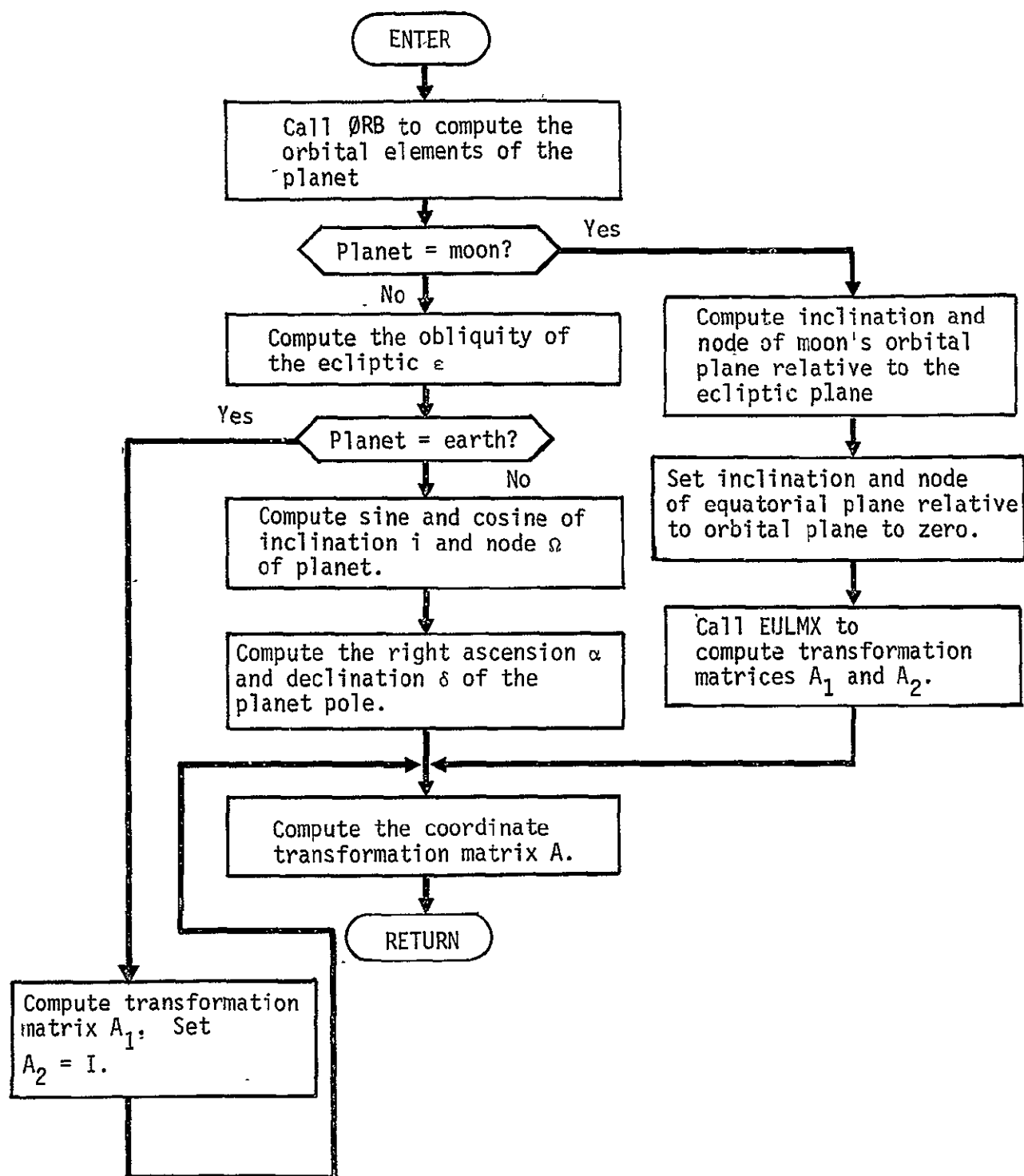
The transformation A_1 from ecliptic to orbital plane coordinates is performed by rotating about the z-axis through an angle Ω and then about the resulting x-axis through an angle i . Symbolically,

$$A_1 = (\Omega \text{ about } 3, i \text{ about } 1). \quad (9)$$

The transformation A_2 from orbital plane to equatorial coordinates can be written similarly as

$$A_2 = (\theta \text{ about } -3, \delta \text{ about } -1). \quad (10)$$

PECEQ Flow Chart



SUBROUTINE PERHEL

PURPOSE: TO PROPAGATE A HELIOCENTRIC TRAJECTORY CONSIDERING THE PERTUBATIONS PRODUCED BY BOTH THE LAUNCH AND TARGET BODIES.

CALLING SEQUENCE: CALL PERHEL(GM,HSI,HLTI,HLTF,DELT,HSF)

ARGUMENT: GM(3) I GRAVITATIONAL CONSTANTS OF SUN, LAUNCH AND TARGET PLANETS

HSI(6) I HELIOCENTRIC ECLIPTIC SPACECRAFT STATE (INITIAL)

HLTI(2,3) I INITIAL HELIOCENTRIC STATES OF LAUNCH AND TARGET BODIES

HLTF(2,3) I FINAL HELIOCENTRIC STATES OF LAUNCH AND TARGET BODIES

DELT I TIME INTERVAL OF PROPAGATION

HSF(6) O HELIOCENTRIC ECLIPTIC SPACECRAFT STATE (FINAL)

SUBROUTINES SUPPORTED: PULCOV PULSEX

SUBROUTINES REQUIRED: BATCON

LOCAL SYMBOLS: COM INTERMEDIATE VARIABLE

DELR RF-RI

PER PERTURBATION IN FINAL STATE

PSF SPACECRAFT POSITION RELATIVE TO PLANET (FINAL)

PSI SPACECRAFT POSITION RELATIVE TO PLANET (INITIAL)

RAV AVERAGE OF RI AND RF

RA INTERMEDIATE VARIABLE

RF MAGNITUDE OF PSF

RH INTERMEDIATE VARIABLE

RI MAGNITUDE OF PSI

PRECEDING PAGE BLANK NOT FILMED

PERHEL Analysis

PERHEL is responsible for propagating a heliocentric trajectory considering the perturbations produced by both the launch and target bodies. The equations of motion of a body moving under the influence of the sun while perturbed by a smaller mass are

$$\ddot{\vec{r}} = -\frac{\mu_0 \vec{r}}{r^3} - \frac{\mu(\vec{r} - \vec{r}_m)}{|\vec{r} - \vec{r}_m|^3} - \frac{\mu \vec{r}_m}{r_m^3} \quad (1)$$

where \vec{r} is the vector radius from the sun to the spacecraft
 \vec{r}_m is the vector radius from the sun to the perturbative mass
 μ_0, μ are the gravitational constants of the sun and mass respectively.

Assuming that the indirect term is small, attention may be directed to the first two terms only. Suppose that $(\vec{r}_0(t), \vec{v}_0(t))$ satisfy

$$\begin{aligned} \dot{\vec{r}}_0 &= \vec{v}_0 \\ \dot{\vec{v}}_0 &= -\frac{\mu_0 \vec{r}_0}{r_0^3} \end{aligned} \quad (2)$$

Then $(r_0(t), v_0(t))$ are given by the familiar equations of conic motion. A first order corrected solution necessary to account for the direct term force must then satisfy

$$\begin{aligned} \dot{\vec{r}} &= \dot{\vec{r}}_0 + \dot{\delta \vec{r}} = \vec{v} \\ \dot{\vec{v}} &= \dot{\vec{v}}_0 + \dot{\delta \vec{v}} = -\frac{\mu_0 \vec{r}_0}{r_0^3} - \mu \frac{(\vec{r}_0 - \vec{r}_m)}{|\vec{r}_0 - \vec{r}_m|^3} \end{aligned} \quad (3)$$

Applying the conditions (2) leads to the equations defining the corrections

$$\begin{aligned} \dot{\delta \vec{r}} &= \delta \vec{v} \\ \dot{\delta \vec{v}} &= -\mu \frac{\vec{R}}{R^3} \end{aligned} \quad (4)$$

where $\vec{R} = \vec{r}_0(t) - \vec{r}_m(t)$ is the position vector of the spacecraft with respect to the perturbing mass.

One further assumption enables one to solve in closed form the perturbations produced by the third mass. Generally $\vec{R}(t)$ and $R(t)$ are nearly linear functions of time. Therefore suppose that the initial and final values of these variables are known to be $\vec{R}_1, \vec{R}_2, R_1, R_2$ over the interval Δt .

Introduce the definitions

$$\begin{aligned}
 \vec{\Delta R} &= \vec{R}_2 - \vec{R}_1 \\
 \Delta R &= R_2 - R_1 \quad (\text{not } |\vec{\Delta R}|). \\
 \langle R \rangle &= \frac{1}{2} (R_1 + R_2) \\
 \Delta R &= \frac{\vec{R}_2}{R_2} - \frac{\vec{R}_1}{R_1}
 \end{aligned} \tag{5}$$

Then the equation defining the velocity perturbation would be

$$\begin{aligned}
 \frac{d\vec{v}}{dt} &= -\mu \frac{\vec{a} + \vec{b} t}{(c + d t)^3} & \vec{a} &= \vec{R}_1 & c &= R_1 \\
 & & \vec{b} &= \frac{\vec{\Delta R}}{\Delta t} & d &= \frac{\Delta R}{\Delta t}
 \end{aligned} \tag{6}$$

It is more convenient however to transform from time t to position magnitude ρ as the independent variable. This may be done since the position magnitude is assumed to be linear in time with $\dot{\rho} = \frac{\Delta R}{\Delta t}$.

According to the assumptions, the position vector \vec{R} is a linear function of ρ also

$$\vec{R} = \vec{A} + \vec{B} \rho \tag{7}$$

Since $\vec{R}(\rho_1) = \vec{R}_1$ and $\vec{R}(\rho_2) = \vec{R}_2$, the constants are

$$\begin{aligned}
 \vec{A} &= \vec{R}_1 - \frac{\vec{\Delta R}}{\Delta R} R_1 = -\frac{R_1 R_2}{\Delta R} \Delta R \\
 \vec{B} &= \frac{\vec{\Delta R}}{\Delta R}
 \end{aligned} \tag{8}$$

In terms of ρ the equations defining the perturbations may be written (with primes indicating differentiation with respect to ρ)

$$\begin{aligned}
 \frac{d\vec{r}'}{d\rho} &= \frac{\Delta t}{\Delta R} \frac{d\vec{v}}{dt} \\
 \frac{d\vec{v}'}{d\rho} &= -\frac{\mu \Delta t}{\Delta R} \frac{\vec{A} + \vec{B} \rho}{\rho^3}
 \end{aligned} \tag{9}$$

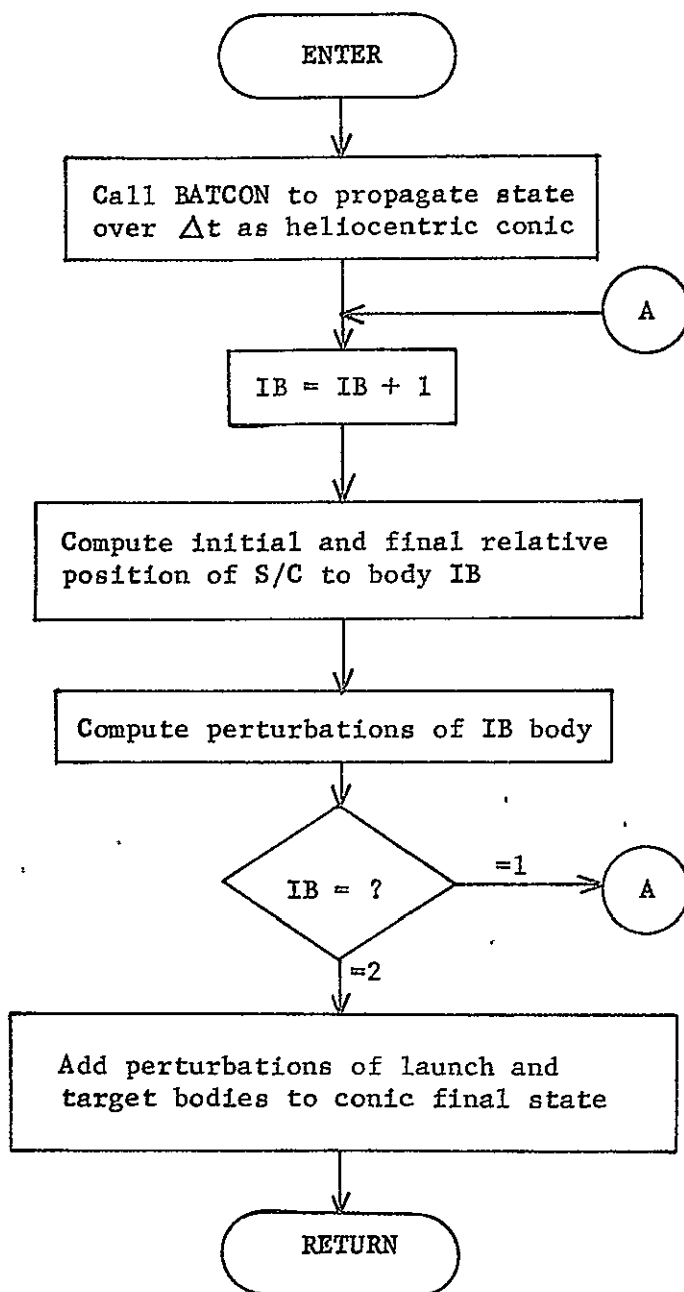
These equations are easily integrated to determine the perturbations caused as the spacecraft moves from \vec{R}_1 to \vec{R}_2 relative to the perturbative body:

$$\begin{aligned}
 \vec{\delta v} &= - \frac{\mu \Delta t}{\Delta R} \int_{R_1}^{\rho} \frac{\vec{A} + \vec{B} \rho}{\rho^3} d\rho \\
 &= \frac{\mu \Delta t}{\Delta R} \left[\frac{\vec{A}}{2} \left(\frac{1}{\rho^2} - \frac{1}{R_1^2} \right) + \vec{B} \left(\frac{1}{\rho} - \frac{1}{R_1} \right) \right] \\
 &= \frac{\mu \Delta t}{R_1 R_2 \Delta R} \left[\langle R \rangle \hat{\Delta R} - \vec{\Delta R} \right], \quad \rho = R_2 \quad (10)
 \end{aligned}$$

$$\begin{aligned}
 \vec{\delta r} &= \frac{\mu \Delta t^2}{\Delta R^2} \int_{R_1}^{R_2} \left[\frac{\vec{A}}{2} \left(\frac{1}{\rho^2} - \frac{1}{R_1^2} \right) + \vec{B} \left(\frac{1}{\rho} - \frac{1}{R_1} \right) \right] d\rho \\
 &= \frac{\mu \Delta t^2}{\Delta R} \left[\frac{1}{2} \frac{\hat{\Delta R}}{R_1} + \frac{\vec{\Delta R}}{\Delta R^2} \left(\ln \left(\frac{R_2}{R_1} \right) - \frac{\Delta R}{R_1} \right) \right] \quad (11)
 \end{aligned}$$

PERHEL calls BATCON for the generation of the uncorrected heliocentric conic, computes the initial and final positions of the spacecraft relative to each of the launch and target planets, and computes the perturbations based on equations (10) and (11) above.

PERHEL Flow Chart



SUBROUTINE PLND

PURPOSE: TO COMPUTE COLUMNS OF THE STATE TRANSITION MATRIX PARTITIONS TXXS, TXW AND TXU ASSOCIATED WITH TARGET PLANET PLANET EPHEMERIS BIASES INCLUDED IN THE AUGMENTED STATE VECTOR BY A NUMERICAL DIFFERENCING TECHNIQUE.

CALLING SEQUENCE: CALL PLND(RI,RF)

ARGUMENT: RF I POSITION AND VELOCITY OF THE VEHICLE AT THE END OF THE TIME INTERVAL

RI I POSITION AND VELOCITY OF THE VEHICLE AT THE BEGINNING OF THE TIME INTERVAL

SUBROUTINES SUPPORTED: PSIM

SUBROUTINES REQUIRED: NTM

LOCAL SYMBOLS: DEL TEMPORARY STORAGE FOR TARGET PLANET SEMI-MAJOR AXIS FACTOR USED IN NUMERICAL DIFFERENCING

IC COUNTER FOR VARIABLES AUGMENTED TO STATE VECTOR

IEMN VECTOR OF INDICES FOR ORBITAL ELEMENTS OF THE MOON

IEND FLAG FOR VARIABLES AUGMENTED TO STATE VECTOR

NEW VECTOR OF INDICES FOR ORBITAL ELEMENTS OF INNER AND OUTER PLANETS

RPER ALTERED FINAL POSITION AND VELOCITY OF VEHICLE

SAVE1 TEMPORARY STORAGE FOR CONSTANTS OF AUGMENTED ELEMENTS OF TARGET PLANET

SAVE2 SAME COMMENTS AS SAVE1

SAVE3 SAME COMMENTS AS SAVE1

COMMON COMPUTED/USED: CN EMN IPRINT SMJR ST

COMMON COMPUTED: TXU TXXS TXW

COMMON USED: ALNGTH DELAXS DELX IAUGDC IAUGH
IAUG NTMC NTP

PLND Analysis

The nonlinear equations of motion of the spacecraft can be written symbolically as

$$\dot{\vec{x}} = \vec{f}(\vec{x}, \vec{e}(t), t) \quad (1)$$

where \vec{x} is the spacecraft position/velocity state and $\vec{e}(t)$ is a vector composed of the six orbital elements a , e , i , Ω , ω , and M of the target planet. The motion of the spacecraft is, of course, dependent on the positions of other celestial bodies, but this dependency need not be explicitly stated for the purposes of this analysis.

Suppose we wish to use numerical differencing to compute those columns of θ_{xx_s} , θ_{xu} , and θ_{xw} associated with target planet ephemeris biases included in the augmented state vector over the time interval $[t_{k-1}, t_k]$. Let $\vec{\theta}_j(t_k, t_{k-1})$ represent the column associated with the j -th ephemeris bias. We assume we have available the nominal states $\vec{x}^*(t_{k-1})$ and $\vec{x}^*(t_k)$, which, of course, were obtained by numerically solving equation (1) using nominal $\vec{e}(t)$. To obtain $\vec{\theta}_j(t_k, t_{k-1})$, we increment the j -th orbital element by the pertinent numerical differencing factor Δe_j and numerically integrate equation (1) over the interval $[t_{k-1}, t_k]$ to obtain the new spacecraft state $\vec{x}_j(t_k)$, where the j -subscript on the spacecraft state indicates that it was obtained by incrementing the j -th orbital element. Then

$$\vec{\theta}_j(t_k, t_{k-1}) = \frac{\vec{x}_j(t_k) - \vec{x}^*(t_k)}{\Delta e_j} \quad (2)$$

Ephemeris biases are defined as biases of the basic set of orbital elements a , e , i , Ω , ω , and M . However the ephemeris constants of a , e , i , Ω , $\tilde{\omega}$, and M for the planets and a , e , i , Ω , $\tilde{\omega}$, and L for the moon are stored in the program. Thus to increment certain of the basic elements, we must increment certain combinations of the stored ephemeris constants.

The elements ω and M are related to the longitude of perihelion $\tilde{\omega}$ and the mean longitude L as follows:

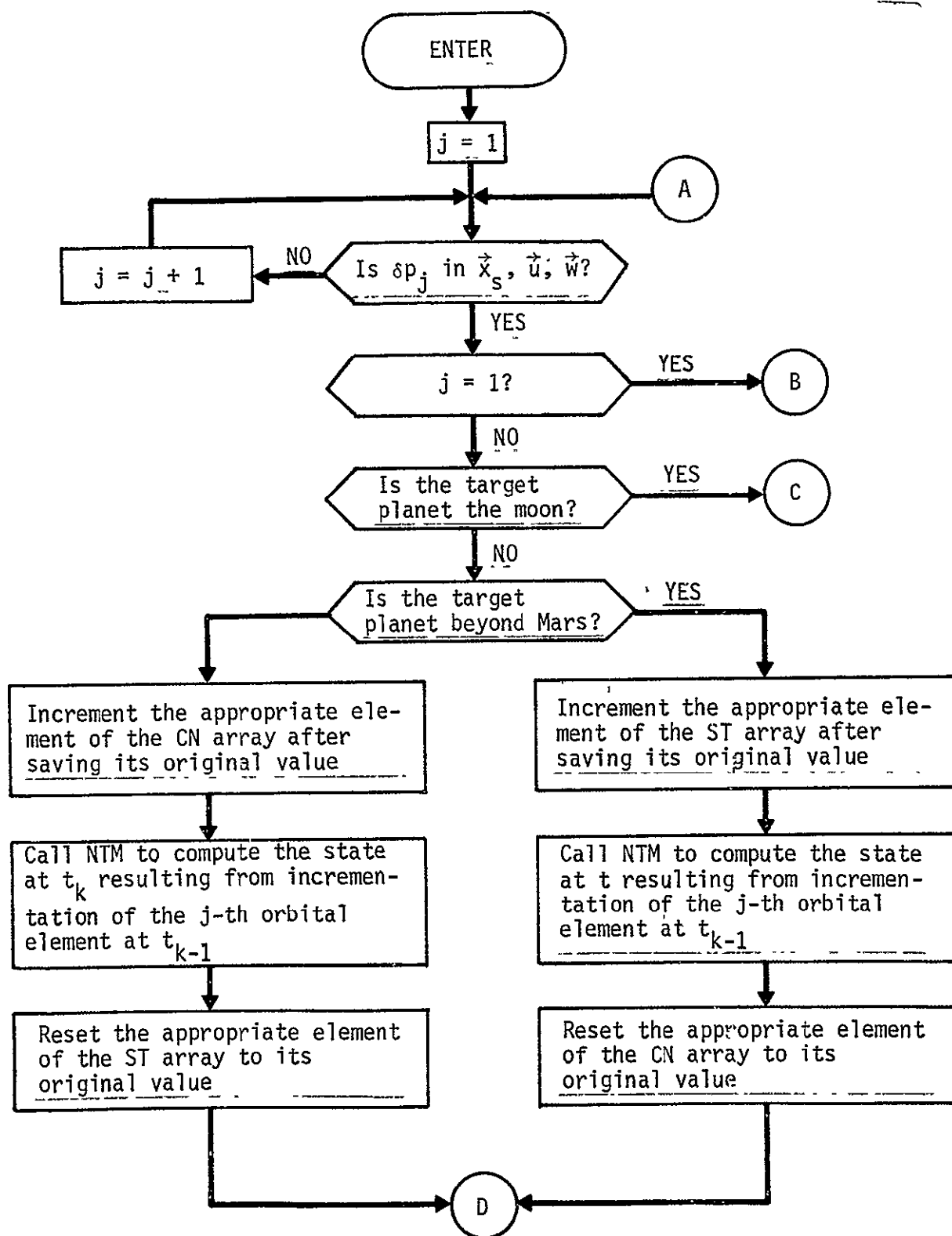
$$\omega = \tilde{\omega} - \Omega$$

$$M = L - \tilde{\omega} \quad .$$

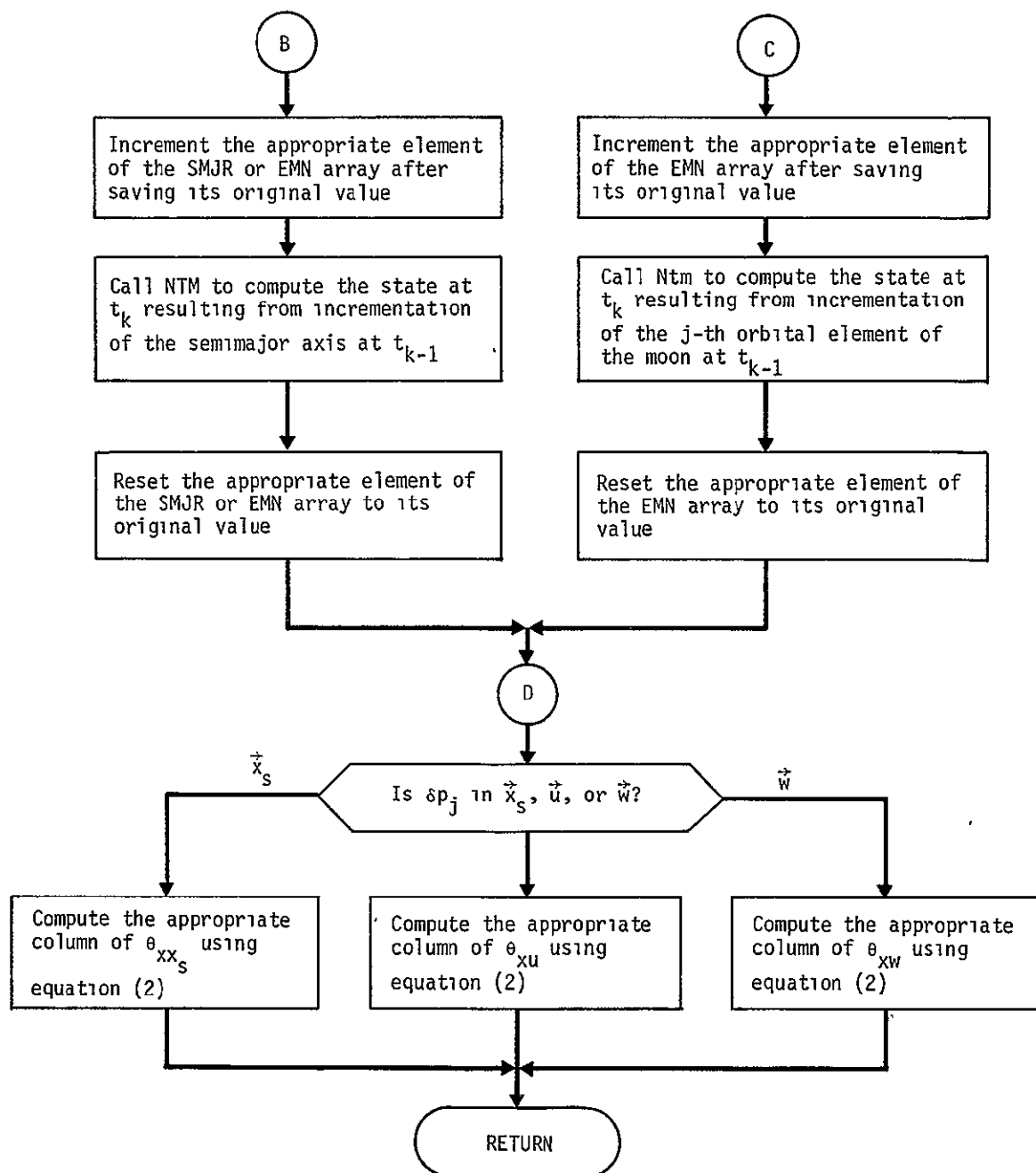
Thus, to increment Ω by $\Delta\Omega$ without changing the other five basic elements requires that we also increment $\tilde{\omega}$ by $\Delta\Omega$ for the case of a planet, and both $\tilde{\omega}$ and L by $\Delta\Omega$ for the case of the moon. To increment ω by $\Delta\omega$ we simply increment $\tilde{\omega}$ by $\Delta\omega$ for a planet, while for the moon we must increment both $\tilde{\omega}$ and L by $\Delta\omega$. To increment M by ΔM for the moon we simply increment L by ΔM .

In the PLND flow chart we employ the following definition:

$$P_j = \left\{ \begin{array}{ll} a & j = 1 \\ e & 2 \\ i & 3 \\ \Omega & 4 \\ \omega & 5 \\ M & 6 \end{array} \right. \quad .$$



PLND Flow Chart



SUBROUTINE POICOM

PURPOSE COMPUTE PROBABILITY OF IMPACT

CALLING SEQUENCE: CALL POICOM(XXXX,DET)

ARGUMENT: XXXX I AIMPOINT IN THE IMPACT PLANE VECTOR

DET I DETERMINANT OF LAMBDA MATRIX

SUBROUTINES SUPPORTED: BIAIM

SUBROUTINES REQUIRED: MATIN

LOCAL SYMBOLS: PMQM P+ MQM TRANSPOSE

SAVE INTERMEDIATE VARIABLE

SUM INTERMEDIATE VARIABLE

W ADA* PMQM* ADA TRANSPOSE

COMMON COMPUTED/USED: IEND POI PSTAR XLAM

COMMON USED:	ADA	A	CR	EXEC	IIGP
	ONE	PI	PP	TWO	XLAMI
	ZERO				

POICOM Analysis

Subroutine POICOM computes the target condition covariance W_j^+ after a guidance correction, the projection of W_j^+ into the impact plane, and the probability of impact of the spacecraft with the target planet.

The target condition covariance matrix W_j^+ is defined as

$$W_j^+ = \eta_j (P_{k_j}^- + M \tilde{Q}_j M^T) \eta_j^T$$

where η_j is the variation matrix for the appropriate guidance policy, $P_{k_j}^-$ is the knowledge covariance prior to the guidance correction, \tilde{Q}_j is the execution error covariance, and M is defined as the following 6×3 matrix:

$$M = \begin{bmatrix} 0 & 1 & 1 \end{bmatrix}^T$$

Before the probability of impact can be computed, it is necessary to compute the projection Λ_j of W_j^+ into the impact plane. The covariance Λ_j is computed as follows for each of the three available midcourse guidance policies.

a. Fixed-time-of-arrival:

$$\Lambda_j = A W_j^+ A^T$$

where transformation A is defined in the subroutine BIAIM analysis.

b. Two-variable B-plane:

$$\Lambda_j = W_j^+$$

c. Three-variable B-plane:

$$\Lambda_j = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} W_j^+ \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}$$

Assuming the probability density function associated with Λ_j is Gaussian and nearly constant over the target planet capture area permits us to compute the probability of impact using the equation

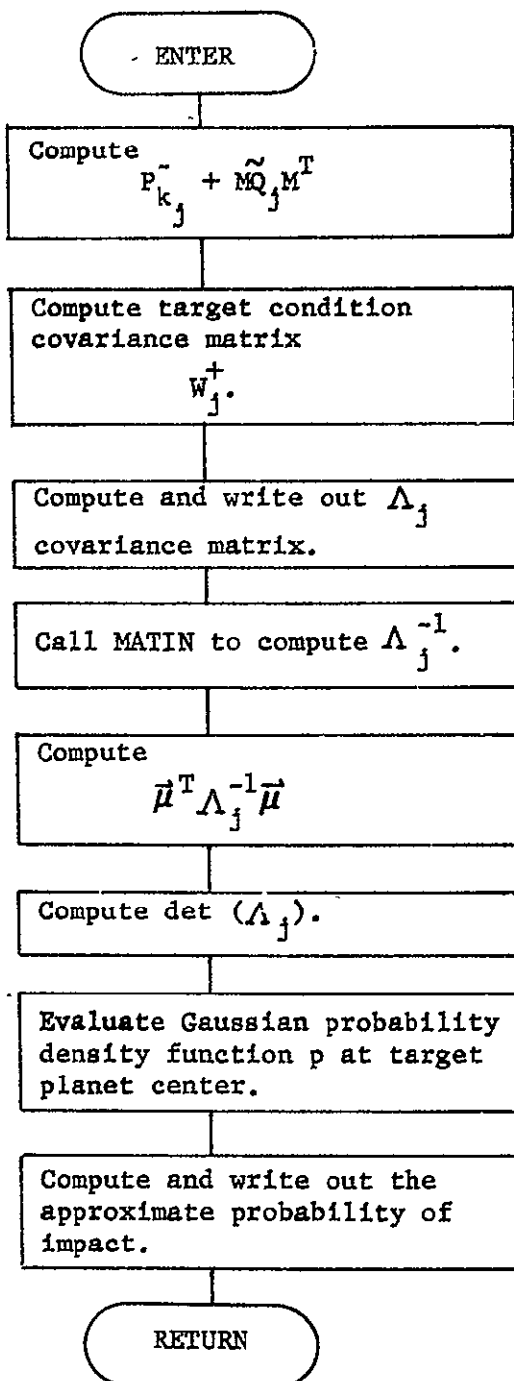
$$POI = \pi R_c^2 p$$

where R_c is the target planet capture radius and p is the Gaussian probability density function evaluated at the target planet center and given by

$$p = \frac{1}{2\pi|\Lambda_j|^{1/2}} \exp \left[-\frac{1}{2} \vec{\mu}^T \Lambda_j^{-1} \vec{\mu} \right]$$

where $\vec{\mu}$ is the aimpoint in the impact plane.

POICOM Flow Chart



PROGRAM PREG

PURPOSE CONTROL EXECUTION OF A PREDICTION EVENT IN THE ERROR
ANALYSIS PROGRAM

SUBROUTINES SUPPORTED: ERRANN

SUBROUTINES REQUIRED:	CORREL	DYNO	HYELS	JACOBI	GNAVM
	NTM	PSIM	STMPR	SAVMAT	EIGHY
	MEAN	GPRINT	ORB	EPHEM	BEPS

LOCAL SYMBOLS:	BLAB	LABEL
	BPS	B-PLANE PARAMETER COVARIANCE
	CXSU1	STORAGE FOR CXSU COVARIANCE ARRAY
	CXSV1	STORAGE FOR CXSV COVARIANCE ARRAY
	CXU1	STORAGE FOR CXU COVARIANCE ARRAY
	CXV1	STORAGE FOR CXV COVARIANCE ARRAY
	CXXS1	STORAGE FOR CXXS COVARIANCE ARRAY
	D	JULIAN DATE PREDICTED TO
	DUMM	INTERMEDIATE VARIABLE
	DUM2	ARRAY OF EIGENVECTORS
	DUM3	ARRAY OF EIGENVALUES
	DUM	B DOT T AND B DOT R COVARIANCE MATRIX
	EGVCT	ARRAY OF EIGENVECTORS
	EGVL	ARRAY OF EIGENVALUES,
	EXSTS	TEMPORARY STORAGE FOR EXST
	EXTS	TEMPORARY STORAGE FOR EXT
	EXTIJ	INTERMEDIATE STORAGE
	ICODE	INTERNAL CONTROL FLAG
	IGO	INTERNAL FLAG
	IPR	STORAGE FOR IPRINT
	OUT	ARRAY OF STANDARD DEVIATIONS AND CORRELATION COEFFICIENTS

PEIG MATRIX WHOSE HYPERELLIPSOID IS TO BE
 COMPUTED

PSAVE TEMPORARY STORAGE FOR P

PSUBB B-PLANE PARAMETER UNCERTAINTIES

PS1 STORAGE FOR PS COVARIANCE ARRAY

P1 STORAGE FOR P COVARIANCE ARRAY

RF NOMINAL SPACECRAFT STATE AT TIME TPT

RFMAG VELOCITY MAGNITUDE

ROW INTERMEDIATE VARIABLE

RPC PLANETOCENTRIC STATE

RPCSV TEMPORARY STORAGE FOR RPC

SQP INTERMEDIATE VARIABLE

SUM INTERMEDIATE VARIABLE

THETA ANGLE

TPT TIME TO WHICH PREDICTION IS TO BE MADE

TRANSG STATE TRANSITION MATRIX RELATING CHANGE IN
 POSITION/VELOCITY TO B-PLANE PARAMETERS

VEIG INTERMEDIATE VECTOR

COMMON COMPUTED/USED:	CXSU	CXSV	CXU	CXV	CXXS
	IPRINT	NPE	PS	P	
	GCXSW	GCXW			
COMMON COMPUTED:	DELT	TRTM1	XI		
COMMON USED:	EM	FOP	FOV	IEIG	IHYP1
	ISTMC	NDIM1	NDIM2	NDIM3	NGE
	NTMC	ONE	Q	TPT2	TSOI1
	GP,	EXT	EXST		
	U0	V0	XF		

PRED Analysis

Subroutine PRED executes a prediction event in the error analysis/generalized covariance analysis program. Subroutine PRED differs from subroutine PRESIM in three respects. First, the propagated knowledge covariance matrix partitions are based on the (most recent) targeted nominal, rather than on the most recent nominal as in PRESIM. Second, estimated position/velocity deviations are not propagated in PRED since estimates are processed only in the simulation program and not in the error analysis program. And third, subroutine PRED treats both assumed and actual knowledge covariance matrix partitions, whereas subroutine PRESIM treats only assumed knowledge covariance matrix partitions. Subroutine PRED uses the propagation equations in subroutine GNAVM to propagate both assumed and actual covariances.

A flow chart for PRED is not presented here because of its similarity to the PRESIM flow chart (see PRESIM for further details).

SUBROUTINE PRELIM

PURPOSE TO PERFORM THE PRELIMINARY WORK ASSOCIATED WITH THE
NOMNAL PROGRAM INCLUDING THE READING OF THE INPUT DATA,
INITIALIZATION OF CONSTANTS, AND THE COMPUTATION OF A
ZERO ITERATE IF REQUIRED

CALLING SEQUENCE: CALL PRELIM

SUBROUTINES SUPPORTED: NOMNAL

SUBROUTINES REQUIRED: CPWMS TIME ZERIT

LOCAL SYMBOLS: DF JULIAN DATE CORRESPONDING TO KALF ARRAY
DI JULIAN DATE CORRESPONDING TO KALI ARRAY
GS ARRAY OF VALUES OF SECONDS CORRESPONDING
TO KALG ARRAY
I INDEX
J INDEX
KALF CALENDAR DATE OF FINAL TRAJECTORY TIME
KALG ARRAY OF CALENDAR DATES OF GUIDANCE EVENTS
KALI CALENDAR DATE OF INITIAL TRAJECTORY TIME
KALT ARRAY OF CALENDAR DATES OF TARGET TIMES
KEY LOCAL VARIABLE USED TO COMPLETE
INFORMATION IN THE ARRAY
SF SECONDS OF FINAL TRAJECTORY TIME
SI SECONDS OF INITIAL TRAJECTORY TIME
TS SECONDS OF TARGET TIMES CORRESPONDING TO
KALT ARRAY

COMMON COMPUTED/USED:	AC	ALNGTH	DG	D1	FI
	IBADS	IBARY	ICoord	IFINT	IPRE
	ISTART	IZERO	KGYD	KMXQ	KOAST
	KTIM	KTYP	LTARG	LVLS	MAT
	MAXB	MDL	NBOD	NB	NCPR
	NOGYD	NOIT	NPAR	ONE	PERV
	PHILS	PSI1	PSI2	RIN	RPRAT
	RP	SIGMAL	SPHFAC	SSS	TAR
	THEDOT	THELS	TIMG	TIM1	TIM2
	TIN	TMPR	TM	ZDAT	

COMMON COMPUTED:

DINTG	EIGHT	FIVE	FOUR	HALF
IEPHEM	IPRINT	KSICA	KUR	NBODYI
NINETY	RAD	TEN	THREE	TMU
TRTM	TWO	ZERO		

COMMON USED:

ACKT	DELV	DT	DVMAX	IBAST
KTAR	LEVELS	MAXBAD	NITS	NLP
NTP	PHI	PMASS	TIMS	TOL

PRELIM Analysis

PRELIM is responsible for the preliminary work required by NOMNAL including the initialization of variables, the reading of input, and the computation of zero iterate values for initial time, position, and velocity if necessary.

On the first call to PRELIM, PRELIM presets constants to be used on the entire series of runs. These constants include the double precision numbers and the launch profile parameters. On subsequent calls these variables are not reset.

PRELIM then presets constants for individual runs. These constants presently include most of the guidance event parameters. The user may easily change the two sets of constants for his particular needs.

PRELIM then accepts the input data. It reads data in the NAMELIST format.

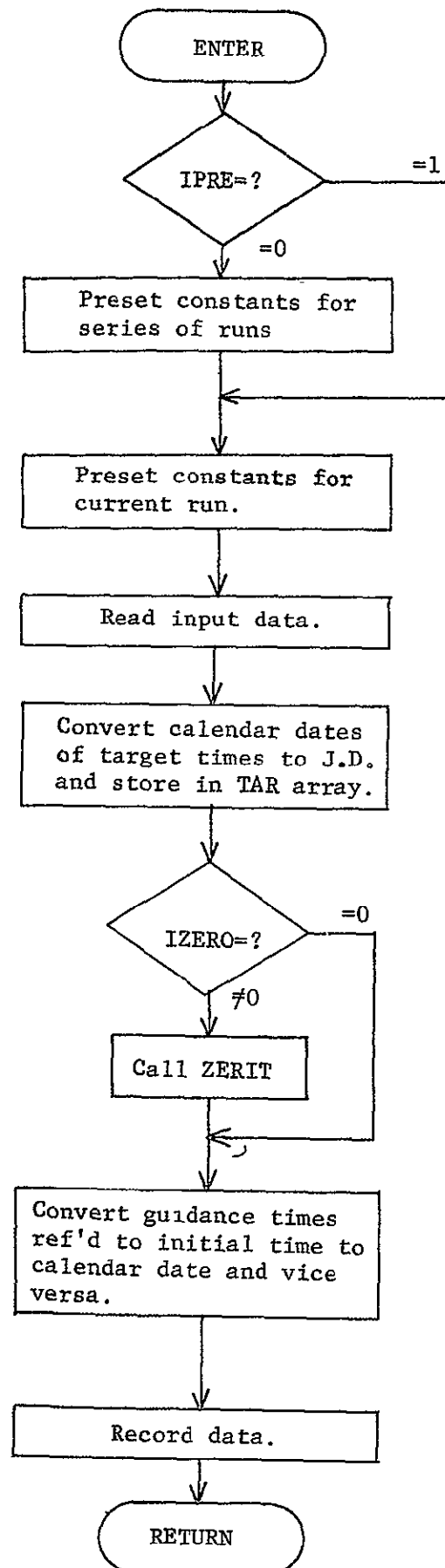
Target times must be read in as calendar dates. PRELIM next converts these to Julian date referenced 1900 and stores the converted values in the TAR array.

If the flag IZERO is nonzero, ZERIT is called for the computation of the zero iterate values of initial time, position, and velocity. ZERIT in turn calls HELIO for interplanetary trajectories and LUNA for lunar trajectories.

PRELIM then converts guidance event times referenced to initial time to calendar data and converts times read in as calendar dates to times referenced to the initial time. When the latter is done, it sets-KTIM to acknowledge that conversion.

Finally PRELIM records all pertinent data.

PRELIM Flow Chart



SUBROUTINE PREPUL

PURPOSE: TO PERFORM THE PRELIMINARY COMPUTATIONS REQUIRED FOR THE PULSING ARC MODEL.

CALLING SEQUENCE: CALL PREPUL(RIN,DELTAV,D1)

ARGUMENTS: RIN(6) I INERTIAL STATE OF SPACECRAFT AT NOMINAL TIME OF CORRECTION

DELTAV(3) I TOTAL VELOCITY INCREMENT TO BE ADDED

D1 I JULIAN DATE OF NOMINAL TIME OF CORRECTION

SUBROUTINES SUPPORTED: EXCUTE EXCUTS

SUBROUTINES REQUIRED: TIME

LOCAL SYMBOLS: A SEMIMAJOR AXIS

C INTERMEDIATE VARIABLE IN F AND G SERIES

DB JULIAN DATE AT BEGINNING OF PULSING ARC

DELVM MAGNITUDE OF TOTAL IMPULSIVE CORRECTION

DE JULIAN DATE AT END OF PULSING ARC

DVFM MAGNITUDE OF FINAL PULSE OF SEQUENCE

DVIM MAGNITUDE OF TYPICAL PULSE OF SEQUENCE

D INTERMEDIATE VARIABLE IN F AND G SERIES

G GRAVITATIONAL CONSTANT OF BODY UNDER CONSIDERATION

ID CALENDAR DATE OF CRITICAL TIMES FOR OUTPUT

MAXP MAXIMUM NUMBER OF PULSES ALLOWED

NDX ARRAY OF CODES OF LAUNCH AND TARGET BODIES

NX INDEX OF GIVEN PLANET COORDINATES IN F-ARRAY

RD TIME DERIVATIVE OF RADIUS MAGNITUDE OF PLANET

RR MAGNITUDE OF RADIUS

SD SECONDS OF CRITICAL TIMES FOR OUTPUT

VV SPEED OF PLANET

COMMON COMPUTED/USED:

B	DVF	DVI	FS	GG
GS	NPUL	PULT	RK	VK

COMMON USED:

ALNGTH	DTI	DUR	FIVE	FOUR
F	NBOD	NB	NINETY	NLP
NTP	PMASS	PULMAG	PULHAS	THREE
TH	TWO	V		

PREPUL Analysis

PREPUL is responsible for performing the preliminary computations required for the pulsing arc model.

PREPUL first determines the nominal pulsing arc. Let the following definitions be made:

T	magnitude of pulsing engine thrust
m	nominal mass of spacecraft
Δt	duration of single pulse
Δt_i	time interval between pulses
$\vec{\Delta v}$	total velocity increment to be added

The velocity increment imparted by a single pulse is

$$\Delta v_i = \frac{T \Delta t}{m} \quad (1)$$

The number of pulses required is then

$$N_p = \left[\frac{\Delta v}{\Delta v_i} \right] + 1 \quad (2)$$

where $[\cdot]$ denotes the greatest integer function. The magnitude of the final pulse must be set to

$$\Delta v_f = \Delta v - (N_p - 1) \Delta v_i \quad (3)$$

The vector nominal pulse and final pulse are therefore given by

$$\begin{aligned} \vec{\Delta v}_i &= \Delta v_i \frac{\vec{\Delta v}}{\Delta v} \\ \vec{\Delta v}_f &= \Delta v_f \frac{\vec{\Delta v}}{\Delta v} \end{aligned} \quad (4)$$

The duration of the pulsing arc is then given by

$$\Delta T = (N_p - 1) \Delta t_i \quad (5)$$

Later computations require time histories of the position vectors of the launch and target bodies. An efficient means of obtaining this involves the f and g series. Given the state \vec{r}_0, \vec{v}_0 of body moving in a conic section about a central body of gravitational constant μ , the position vector as a function of t measured from the initial time is given by

$$\vec{r}(t) = f(t) \vec{r}_0 + g(t) \vec{v}_0 \quad (6)$$

where

$$f(t) = \sum_{k=0}^n f_k t^k \quad g(t) = \sum_{k=1}^n g_k t^k \quad (7)$$

The constants f_k, g_k are computed in PREPUL as

$$f_0 = 1$$

$$f_1 = 0$$

$$f_2 = \frac{-\mu}{2r_o^3}$$

$$f_3 = \frac{\mu \dot{r}_o}{2r_o^4}$$

$$f_4 = \frac{\mu^2}{24r_o^6} \left(4 - 15 \frac{r_o \dot{r}_o^2}{\mu} - 3 \frac{r_o}{a} \right)$$

$$f_5 = \frac{-\mu^2 \dot{r}_o}{8r_o^7} \left(4 - \frac{7r_o \dot{r}_o^2}{\mu} - 3 \frac{r_o}{a} \right)$$

$$f_6 = \frac{\mu^3}{720 r_o^9} \left[-70 + 114 \frac{r_o}{a} + 840 \frac{r_o \dot{r}_o^2}{\mu} - 630 \frac{r_o^2 \dot{r}_o^2}{\mu a} - 450 \left(\frac{r_o \dot{r}_o^2}{\mu} \right)^2 - 45 \frac{r_o^2}{a^2} \right]$$

$$g_1 = 1$$

$$g_2 = 0$$

$$g_3 = \frac{1}{3} f_2$$

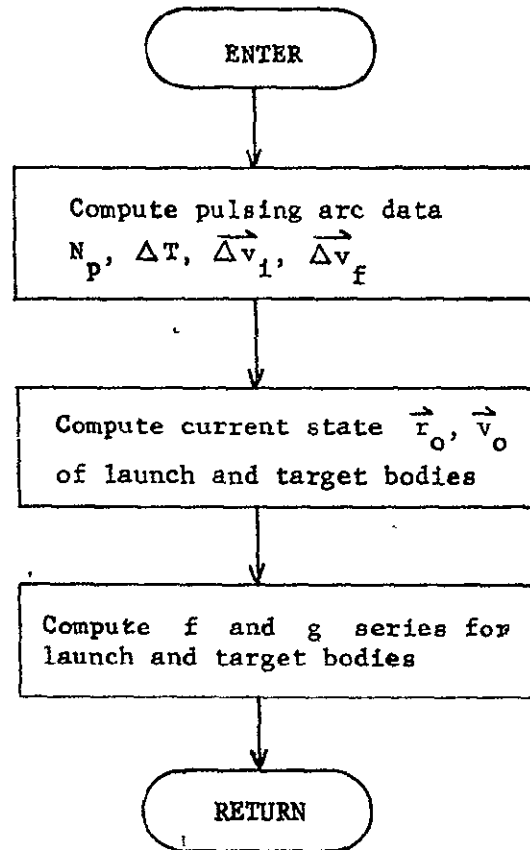
$$g_4 = \frac{1}{2} f_3$$

$$g_5 = \frac{3}{5} f_4 - \frac{1}{15} f_2^2$$

$$g_6 = \frac{2}{3} f_5 - \frac{1}{6} f_2 f_3$$

Reference: Baker, R. M. L. and Makemson, M. W., An Introduction to Astrodynamics, Academic Press, New York, 1967.

PREPUL Flow Chart



PROGRAM PRESIM

PURPOSE CONTROL EXECUTION OF A PREDICTION EVENT IN THE
SIMULATION PROGRAM

SUBROUTINES SUPPORTED: SIMULL

SUBROUTINES REQUIRED:	CORREL	DYNOS	HYELS	JACOBI	NAVM
	NTMS	PSIM	STMPR	EIGHY	ORB
	EPHEM	BEPS			

LOCAL SYMBOLS:	BLAB	LABEL
	BPS	B-PLANE PARAMETER COVARIANCE
	CXSU1	STORAGE FOR CXSU COVARIANCE ARRAY
	CXSV1	STORAGE FOR CXSV COVARIANCE ARRAY
	CXU1	STORAGE FOR CXU COVARIANCE ARRAY
	CXV1	STORAGE FOR CXV COVARIANCE ARRAY
	CXXS1	STORAGE FOR CXXS COVARIANCE ARRAY
	D	JULIAN DATE PREDICTED TO
	DM	B DOT T AND B DOT R COVARIANCE MATRIX
	DM2	ARRAY OF EIGENVECTORS
	DM3	ARRAY OF EIGENVALUES
	DUMM	INTERMEDIATE VARIABLE
	EGVCT	ARRAY OF EIGENVECTORS
	EGVL	ARRAY OF EIGENVALUES
	GMU	GRAVITATIONAL CONSTANT OF TARGET PLANET
	IPR	STORAGE FOR IPRINT
	OUT	ARRAY OF STANDARD DEVIATIONS AND CORRELATION COEFFICIENTS
	PEIG	MATRIX WHOSE HYPERELLIPSOID IS TO BE COMPUTED
	PSUBB	B-PLANE PARAMETER UNCERTAINTIES
	PS1	STORAGE FOR PS COVARIANCE ARRAY

P1 STORAGE FOR P COVARIANCE ARRAY
 RFMAG VELOCITY MAGNITUDE
 RF1 MOST RECENT NOMINAL SPACECRAFT STATE AT
 TIME TPT2
 ROW INTERMEDIATE VARIABLE
 RPC PLANETOCENTRIC STATE
 SQP INTERMEDIATE VARIABLE
 SUM INTERMEDIATE VARIABLE
 THETA ANGLE
 TPT TIME TO WHICH PREDICTION IS TO BE MADE
 TRANSG STATE TRANSITION MATRIX RELATING CHANGE IN
 POSITION/VELOCITY TO B-PLANE PARAMETERS
 VEIG MATRIX TO BE DIAGONALIZED

COMMON COMPUTED/USED:	CXSV	CXU	CXV	CXXS	
	ICODE	NPE	PS	P	
	RI1				
COMMON COMPUTED:	DELTM	RI	TRTM1	XI1	
				XI	
COMMON USED:	ADEVXS	ADEVX	EDEVXS	EDEVX	EM
	FOP	FOV	IEIG	IHYP1	ISTMC
	NDIM1	NDIM2	NDIM3	NGE	NTMC
	ONE	PHI	Q	TEVN	TPT2
	TSOI1	TXXS	U0	V0	W
	XF1	XF	XSL	ZERO	

PRESIM Analysis

Subroutine PRESIM executes a prediction event in the simulation program SIMUL. At a prediction event, the knowledge covariance partitions, and the estimated position/velocity deviations from the most recent nominal trajectory are propagated forward to t_p , the time to which the prediction is to be made. The knowledge covariance partitions are propagated using the prediction equations found in the NAVM Analysis section. The estimate is propagated using the equation

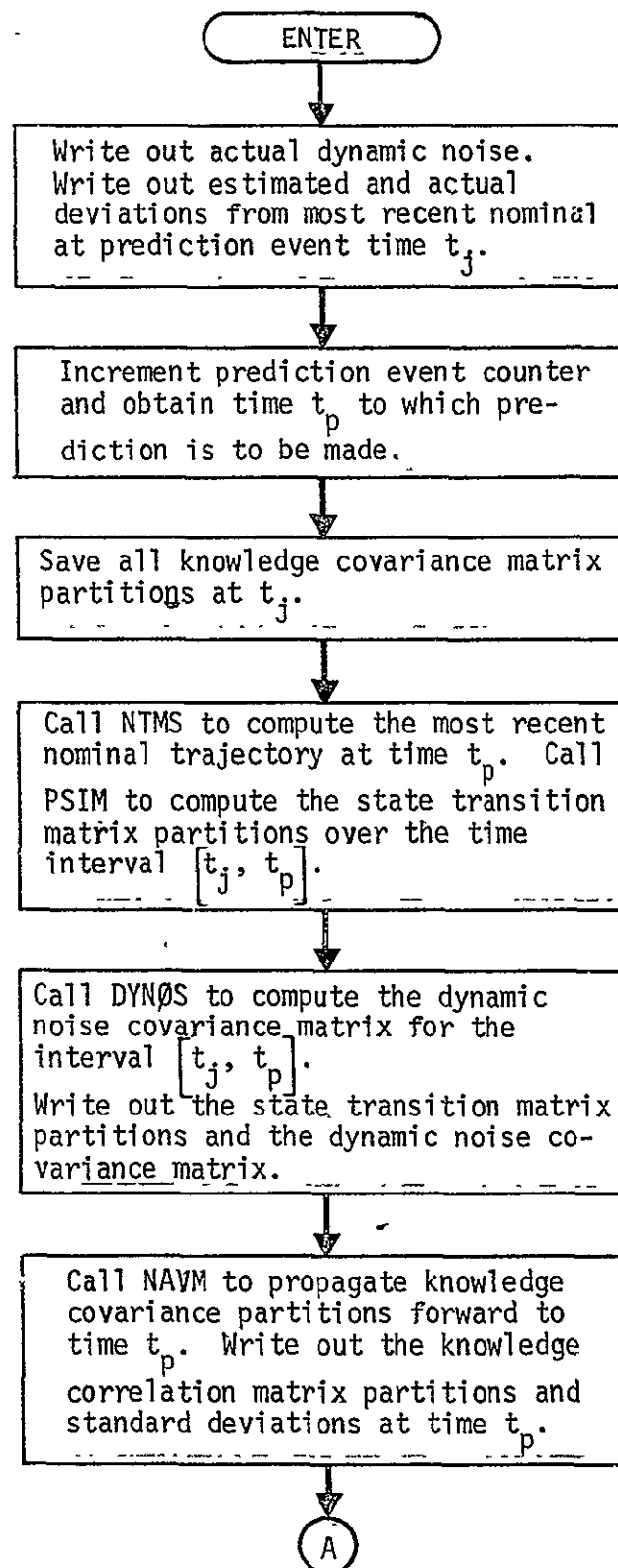
$$\tilde{\mathbf{X}}_p = \Phi(t_p, t) \tilde{\mathbf{X}}_j + \theta_{xx_s}(t_p, t_j) \tilde{\mathbf{X}}_{s_j}$$

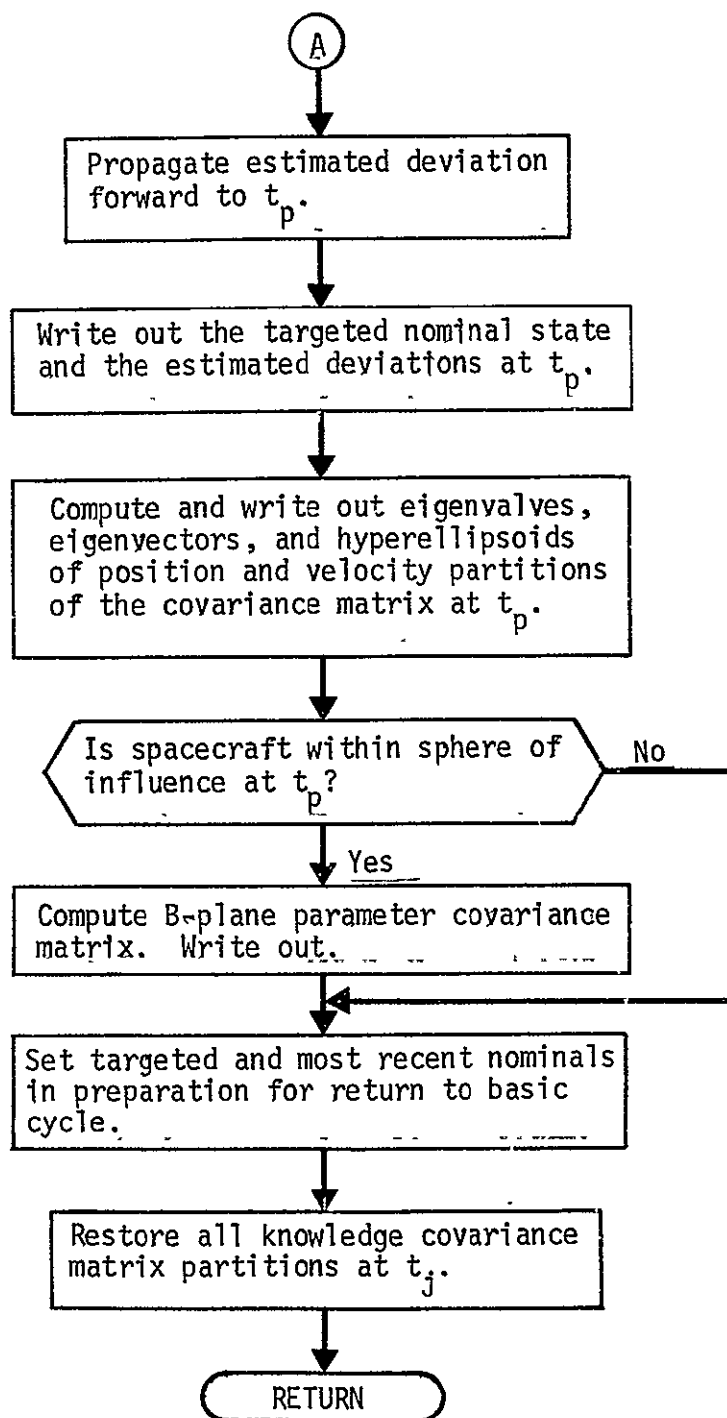
where Φ and θ_{xx_s} are the state transition matrix partitions over the time interval $[t_j, t_p]$.

The position and velocity partitions of the propagated knowledge covariance matrix are diagonalized at time t_p and the eigenvalues, eigenvectors, and hyperellipsoids are computed.

If t_p occurs within the target planet sphere of influence, the Cartesian position/velocity covariance matrix is transformed to a B-plane parameter covariance matrix. The B-plane parameters are B·T, B·R, time-of-flight, S·R, S·T, and C_3 .

PRESIM Flow Chart





SUBROUTINE PRINT

PURPOSE: TO PRINT THE VIRTUAL MASS INFORMATION SPECIFIED.

CALLING SEQUENCE: CALL PRINT

SUBROUTINES SUPPORTED: VMP

SUBROUTINES REQUIRED: TIME TRAPAR NEWPGE SPACE

LOCAL SYMBOLS:	D	INTERMEDIATE VARIABLE USED FOR PRINTOUT
	IDAY	DAY OF CALENDAR DATE OF CURRENT TIME
	IHR	HOUR OF CALENDAR DATE OF CURRENT TIME
	INCMNT	CURRENT TOTAL INCREMENTS FOR PRINTOUT
	IP	CODE OF I-TH PLANET FOR PRINTOUT PURPOSES
	IYR	YEAR OF CALENDAR DATE OF CURRENT TIME
	MIN	MINUTES OF CALENDAR DATE OF CURRENT TIME
	MO	MONTH OF CALENDAR DATE OF CURRENT TIME
	RP	RADIUS OF I-TH PLANET RELATIVE TO INERTIAL FRAME
	RS	RADIUS OF VEHICLE RELATIVE TO INERTIAL FRAME
	RV	RADIUS OF VIRTUAL MASS RELATIVE TO INERTIAL FRAME
	SEC	SECONDS OF CALENDAR DATE OF CURRENT TIME
	TMP	POSITION AND VELOCITY OF VIRTUAL MASS RELATIVE TO PLANETS
	VMR	MAGNITUDE OF VELOCITY OF VEHICLE RELATIVE TO VIRTUAL MASS
	VP	MAGNITUDE OF VELOCITY OF I-TH PLANET FOR PRINTOUT PURPOSES
	VS	MAGNITUDE OF VELOCITY OF VEHICLE RELATIVE TO INERTIAL FRAME
	VSP	MAGNITUDE OF VELOCITY OF VEHICLE RELATIVE TO I-TH PLANET FOR PRINTOUT PURPOSES
	VV	MAGNITUDE OF VELOCITY OF VIRTUAL MASS

0.11

RELATIVE TO INERTIAL FRAME

COMMON COMPUTED/USED:

F V

COMMON USED:

INCMNT	IPRT	NBODYI	NBODY	NO
PLANET	ZERO			

SUBROUTINE PRINT3

PURPOSE: TO PRINT THE PERTINENT INFORMATION AT THE END OF EACH MEASUREMENT.

CALLING SEQUENCE: CALL PRINT3

SUBROUTINES SUPPORTED: ERRANN PROBE

SUBROUTINES REQUIRED: CORREL EPHEM ORB STMPR TRAPAR

LOCAL SYMBOLS:	D	INTERMEDIATE DATE
	IA	STATION NUMBER
	D3	JULIAN DATE OF INITIAL TIME
	D4	JULIAN DATE OF FINAL TIME
	IDAY	CALENDAR DAY OF FINAL TIME
	IHR	CALENDAR HOUR OF FINAL TIME
	IMIN	CALENDAR MINUTE OF FINAL TIME
	IMO	CALENDAR MONTH OF FINAL TIME
	ITEMP	INTERMEDIATE VARIABLE
	IYR	CALENDAR YEAR OF FINAL TIME
	LDAY	CALENDAR DAY OF INITIAL TIME
	LHR	CALENDAR HOUR OF INITIAL TIME
	LMIN	CALENDAR MINUTES OF INITIAL TIME
	LMO	CALENDAR MONTH OF INITIAL TIME
	LYR	CALENDAR YEAR OF INITIAL TIME
	M	NUMBER OF MEASUREMENT
	RME	GEOCENTRIC RADIUS OF VEHICLE
	RMP	DISTANCE OF VEHICLE FROM TARGET PLANET
	SECI	CALENDAR SECONDS OF FINAL TIME
	SECL	CALENDAR SECONDS OF INITIAL TIME
	TRTM2	TRAJECTORY TIME AT END OF INTERVAL

VME MAGNITUDE OF VELOCITY OF VEHICLE RELATIVE TO
TO EARTH

VMP MAGNITUDE OF VELOCITY OF VEHICLE RELATIVE TO
TO TARGET PLANET

COMMON COMPUTED/USED:

COMMON USED:

NO	RE	RTP	XP	
AK	ALNGTH	AL	AM	CXSUP
CXSU	CXSVP	CXSV	CXUP	CXU
CXVP	CXV	CXXSP	CXXS	DATEJ
DELT	F	G	H	IBARY
IPROB	IPRT	MCNTR	NBOD	NB
NDIM1	NDIM2	NDIM3	NTP	PP
PSP	PS	P	Q	R
S	TM	TRTM1	U0	V0
XF	XI	XLAB	XSL	XU
XV	ALPHA	BETA	NDIM4	XIG
AN				

SUBROUTINE PRINT4

PURPOSE: THIS SUBROUTINE PRINTS RELEVANT DATA AT THE END OF EACH MEASUREMENT IN THE SIMULATION MODE

CALLING SEQUENCE: CALL PRINT4(MMCODE,NR)

ARGUMENT: MMCODE I MEASUREMENT CODE

NR I NUMBER OF ROWS IN THE OBSERVATION MATRIX

SUBROUTINES SUPPORTED: SIMULL

SUBROUTINES REQUIRED: CORREL EPHEM ORB STMPR SUB1
TRAPAR

LOCAL SYMBOLS: ADON ACTUAL STATE DEVIATION FROM TARGETED
NOMINAL TRAJECTORY

AODI ACTUAL ORBIT ESTIMATION ERROR

D INTERMEDIATE DATE

EDON ESTIMATED STATE DEVIATION FROM TARGETED
NOMINAL TRAJECTORY

IA STATION NUMBER

IB STAR#PLANET ANGLE NUMBER

M MEASUREMENT NUMBER

ROW ARRAY OF CORRELATION COEFFICIENTS

SQP VECTOR OF STANDARD DEVIATIONS

TRTM2 TRAJECTORY TIME AT END OF INTERVAL

XE1 POSITION AND VELOCITY OF EARTH AT TRTM1

XE2 POSITION AND VELOCITY OF EARTH AT TRTM2

XP1 POSITION AND VELOCITY OF TARGET PLANET AT
TRTM1

XP2 POSITION AND VELOCITY OF TARGET PLANET AT
TRTM2

COMMON COMPUTED/USED: NO

COMMON USED: ADEVXS ADEV# AK ALNGTH AL
AN ANOIS AR AY CXSUP

CXSU	CXSVP	CXSV	CXUP	CXU
CXVP	CXV	CXXSP	CXXS	DATEJ
DELT	EDEVXS	EDEVX	EY	F
G	HPR	H	IBARY	IPROB
IPRT	MCNTR	NBOD	NB	NDIM1
NDIM2	NDIM3	NTP	PP	PSP
PS	P	Q	RES	R
S	TM	TRTM1	U0	V0
W	XF1	XF	XI1	XI
XLAB	XP	XSL	XU	XV
ZF	ZI			

PROGRAM PRNTS3

PURPOSE: TO PRINT A SUMMARY OF THE ERROR ANALYSIS MODE

SUBROUTINES SUPPORTED: ERRON

SUBROUTINES REQUIRED: CORREL TIME

LOCAL SYMBOLS: D8 HOLLERITH LABEL INITIAL
D9 HOLLERITH LABEL FINAL
D1 JULIAN DATE, EPOCH JAN.0,1900, OF INITIAL TIME
D2 JULIAN DATE, EPOCH JAN. 0,1900, OF FINAL TIME
D3 JULIAN DATE OF INITIAL TIME
D4 JULIAN DATE OF FINAL TIME
F FUNCTION= SQUARE ROOT OF SUM OF 3 SQUARES
IDAY CALENDAR DAY OF FINAL TIME
IHR CALENDAR HOUR OF FINAL TIME
IMIN CALENDAR MINUTES OF FINAL TIME
IMO CALENDAR MONTH OF FINAL TIME
IYR CALENDAR YEAR OF FINAL TIME
LDAY CALENDAR DAY OF INITIAL TIME
LHR CALENDAR HOUR OF INITIAL TIME
LMIN CALENDAR MINUTES OF INITIAL TIME
LMO CALENDAR MONTH OF INITIAL TIME
LYR CALENDAR YEAR OF INITIAL TIME
RI POSITION AND VELOCITY OF VEHICLE AT INITIAL TIME
RMF HELIOCENTRIC RADIUS OF VEHICLE AT FINAL TIME
RMI HELIOCENTRIC RADIUS OF VEHICLE AT INITIAL TIME

SECI CALENDAR SECONDS OF FINAL TIME
 SECL CALENDAR SECONDS OF INITIAL TIME
 TRTM2 TRAJECTORY TIME AT END OF TRAJECTORY
 VE POSITION AND VELOCITY OF VEHICLE RELATIVE
 TO EARTH AT FINAL TIME
 VMF MAGNITUDE OF VELOCITY OF VEHICLE AT FINAL
 TIME
 VMI MAGNITUDE OF VELOCITY OF VEHICLE AT
 INITIAL TIME
 VT POSITION AND VELOCITY OF VEHICLE RELATIVE TO
 TARGET PLANET AT FINAL TIME

COMMON COMPUTED/USED: DC DSI

COMMON COMPUTED: TRTM1

COMMON USED:

ACCND	ACC	ALNGTH	BDRSI1	BDTSI1
BSI1	B	CXSU	CXSV	CXU
CXV	CXXS	DATEJ	DELTH	DELX
DNCN	DTMAX	FACP	FACV	FNTM
IAUGIN	IDNF	IEPHEM	IMNF	IPROB
ISPH	ISTMC	ISTM1	MNCN	MNNAME
NDACC	NDIM1	NDIM2	NDIM3	NEV1
NEV2	NEV3	NEV	NMN	NST
NTMC	NTP	PB	PLANET	PSB
PS	P	RCA1	RE	RSOI1
RTP	SAL	SIGALP	SIGBET	SIGPRO
SIGRES	SLAT	SLON	TCA1	TM
TRTMB	TSOI1	UST	U0	VSOI1
VST	V0	WST	X8	XNM

PROGRAM PRNTS4

PURPOSE: TO PRINT OUT A SUMMARY OF THE SIMULATION MODE

SUBROUTINES SUPPORTED: SIMULL

SUBROUTINES REQUIRED: CORREL EPHEM ORB TIME
NOMINAL

LOCAL SYMBOLS:

ADON	ACTUAL STATE DEVIATION FROM TARGETED NOMINAL TRAJECTORY AT FINAL TIME
AODI	ACTUAL ORBIT ESTIMATION ERROR AT FINAL TIME
BLANK	BLANK HOLLERITH CHARACTER
D1	JULIAN DATE, EPOCH JAN.0,1900, OF INITIAL TIME
D2	JULIAN DATE, EPOCH JAN.0,1900, OF FINAL TIME
D3	JULIAN DATE OF INITIAL TIME
D4	JULIAN DATE OF FINAL TIME
EDON	ESTIMATED STATE DEVIATION FROM TARGETED NOMINAL TRAJECTORY
IDAY	CALENDAR DAY OF FINAL TIME
IHR	CALENDAR HOUR OF FINAL TIME
IMIN	CALENDAR MINUTES OF FINAL TIME
IMO	CALENDAR MONTH OF FINAL TIME
IYR	CALENDAR YEAR OF FINAL TIME
LDAY	CALENDAR DAY OF INITIAL TIME
LHR	CALENDAR HOURS OF INITIAL TIME
LMIN	CALENDAR MINUTES OF INITIAL TIME
LMO	CALENDAR MONTH OF INITIAL TIME
LYR	CALENDAR YEAR OF INITIAL TIME
RE1	POSITION AND VELOCITY OF VEHICLE RELATIVE TO TO EARTH ON TARGETED NOMINAL

RE2	POSITION AND VELOCITY OF VEHICLE RELATIVE TO TO EARTH ON MOST RECENT NOMINAL
RE3	POSITION AND VELOCITY OF VEHICLE RELATIVE TO TO EARTH ON ACTUAL TRAJECTORY
RME1	GEOCENTRIC RADIUS OF VEHICLE ON TARGETED NOMINAL AT FINAL TIME
RME2	GEOCENTRIC RADIUS OF VEHICLE ON MOST RECENT NOMINAL AT FINAL TIME
RME3	GEOCENTRIC RADIUS OF VEHICLE ON ACTUAL TRAJECTORY AT FINAL TIME
RME	GEOCENTRIC RADIUS OF VEHICLE AT INITIAL TIME
RMP1	DISTANCE OF VEHICLE FROM TARGET PLANET ON TARGETED NOMINAL AT FINAL TIME
RMP2	DISTANCE OF VEHICLE FROM TARGET PLANET ON MOST RECENT NOMINAL AT FINAL TIME
RMP3	DISTANCE OF VEHICLE FROM TARGET PLANET ON ACTUAL TRAJECTORY AT FINAL TIME
RMP	DISTANCE OF VEHICLE FROM TARGET PLANET AT INITIAL TIME
RMS1	HELIOCENTRIC RADIUS OF VEHICLE AT FINAL TIME ON TARGETED NOMINAL
RMS2	HELIOCENTRIC RADIUS OF VEHICLE AT FINAL TIME ON MOST RECENT NOMINAL
RMS3	HELIOCENTRIC RADIUS OF VEHICLE AT FINAL TIME ON ACTUAL TRAJECTORY
RMS	HELIOCENTRIC RADIUS OF VEHICLE AT INITIAL TIME
RP1	STATE OF VEHICLE RELATIVE TO TARGET PLANET AT FINAL TIME ON TARGETED NOMINAL
RP2	STATE OF VEHICLE RELATIVE TO TARGET PLANET AT FINAL TIME ON MOST RECENT NOMINAL
RP3	STATE OF VEHICLE RELATIVE TO TARGET PLANET AT FINAL TIME ON ACTUAL TRAJECTORY
SECI	CALENDAR SECONDS AT FINAL TIME

SECL	CALENDAR SECONDS AT INITIAL TIME
VME	MAGNITUDE OF VELOCITY OF VEHICLE RELATIVE TO EARTH AT INITIAL TIME
VME1	MAGNITUDE OF VELOCITY OF VEHICLE RELATIVE TO EARTH ON TARGETED NOMINAL AT FINAL TIME
VME2	MAGNITUDE OF VELOCITY OF VEHICLE RELATIVE TO EARTH ON MOST RECENT NOMINAL AT FINAL TIME
VME3	MAGNITUDE OF VELOCITY OF VEHICLE RELATIVE TO EARTH ON ACTUAL TRAJECTORY AT FINAL TIME
VMP	MAGNITUDE OF VELOCITY OF VEHICLE RELATIVE TO TARGET PLANET AT INITIAL TIME
VMP1	MAGNITUDE OF VELOCITY OF VEHICLE RELATIVE TO TARGET PLANET ON TARGETED NOMINAL AT FINAL TIME
VMP2	MAGNITUDE OF VELOCITY OF VEHICLE RELATIVE TO TARGET PLANET ON MOST RECENT NOMINAL AT FINAL TIME
VMP3	MAGNITUDE OF VELOCITY OF VEHICLE RELATIVE TO TARGET PLANET ON ACTUAL TRAJECTORY AT FINAL TIME
VMS	MAGNITUDE OF VELOCITY OF VEHICLE AT INITIAL TIME
VMS1	MAGNITUDE OF VELOCITY OF VEHICLE AT FINAL TIME ON TARGETED NOMINAL
VMS2	MAGNITUDE OF VELOCITY OF VEHICLE AT FINAL TIME ON MOST RECENT NOMINAL TRAJECTORY
VMS3	MAGNITUDE OF VELOCITY OF VEHICLE AT FINAL TIME ON ACTUAL TRAJECTORY

COMMON COMPUTED/USED:	NO	RE	RTP	XP	ZI
-----------------------	----	----	-----	----	----

COMMON USED:	AALP	ABET	ACCND	ACC1	ACC
	ADEVXB	ADEVXS	ADEVX	ALNGTH	APRO
	ARES	AVARM	BORSI1	BORSI2	BORSI3
	BDTSI1	BDTSI2	BDTSI3	BSI1	BSI2

BSI3	B	CXSUB	CXSU	CXSVB
CXSV	CXUB	CXU	CXVB	CXV
CXXSB	CXXS	DAB	DATEJ	DEB
DELMUP	DELMUS	DELX	DIB	DMAB
DMUPB	DMUSB	DNCN	DNOB	DTMAX
DWB	EDEVXS	EDEVX	FACP	FACV
FNTM	F	H	IAMNF	IAUGIN
IBARY	IDNF	IMNF	IPROB	ISOI1
ISOI2	ISOI3	ISTMC	ISTM1	MNCN
MNNAME	NBOD1	NBOD	NB1	NB
NDACC	NDIM1	NDIM2	NDIM3	NEV1
NEV2	NEV3	NEV5	NEV	NMN
NTMC	NTP	PB	PLANET	PSB
PS	P	RCA1	RCA2	RCA3
RSOI1	RSOI2	RSOI3	SAL	SIGALP
SIGBET	SIGPRO	SIGRES	SLAT	SLON
TCA1	TCA2	TCA3	TM	TRTMB
TSOI1	TSOI2	TSOI3	TTIM1	TTIM2
UNMAC	UST	U0	VSOI1	VSOI2
VSOI3	VST	V0	WST	XB
XF1	XF	XLAB	XNM	ZF

SUBROUTINE PROBE

PURPOSE: TO CONTROL THE EXECUTION OF ALL PROBE RELEASE EVENTS IN
ERRAN

CALLING SEQUENCE: CALL PROBE

SUBROUTINES SUPPORTED: ERPAN

SUBROUTINES REQUIRED: NTM PSIM DYN0 GNAVM MEAN
 ORB EPHEM STMPR CORREL NTRY
 SCHED TRAKM MENO PRINT3 TPRTRG
 MINIQ

LOCAL SYMBOLS: AI MINI-PROBE ROLL RELEASE ANGLE
 AY DUMMY ARGUMENT FOR TRAKM
 COSA COS(ALFA)
 COSAI COS(AI)
 COSD COS(DELT)
 ICL2S TEMPORARY STORAGE FOR ICL2
 IPRN PRINT COUNTER FOR MEASUREMENT PROCESSING
 ISP2S TEMPORARY STORAGE FOR ISP2
 MAMI =1 IF MAIN PROBE BEING PROCESSED
 =2 IF MINI-PROBE BEING PROCESSED
 MAMIP =MAMI + 1 (INPUT TO SCHED, POINTS TO PROPER
 SCHEDULE OF MEASUREMENTS)
 MCNTRS TEMPORARY STORAGE FOR MCNTR
 NMNS TEMPORARY STORAGE FOR NMN
 NMP NUMBER OF MINI-PROBE BEING PROCESSED
 PI MATHEMATICAL CONSTANT
 RF STATE VECTOR AT END OF TIME INTERVAL
 SINA SIN(ALFA)
 SINAI SIN(AI)
 SIND SIN(DELT)
 SMNCN TEMOPRARY STORAGE FOR MNCN ARRAY

SPHERS TEMOPRARY STORAGE FOR SPHERE (NTP)

TE TIME AT THE PROBE SPHERE

TEVNS TEMPORARY STORAGE FOR TEVN

VT TANGENTIAL VELOCITY OF MINI-PROBE

XFP HELIOCENTRIC STATE AT PROBE SPHERE

XPS PLANETOCENTRIC STATE AT PROBE SPHERE

XSAVE HELIOCENTRIC STATE AT BEGINNING OF PROBE
EVENT

COMMON COMPUTED/USED:	CXSU	CXSUG	CXSV	CXSVG	CXU
	CXUG	CXV	CXVG	CXXS	CXXSG
	DELT	EXI	EXSI	EXST	EXT
	GCXSW	GCXM	GCXUG	GCXVG	GCXW
	GCXWG	GCXXSG	GU	GV	GPG
	GPSG	ICL2	MCNTR	MMCODE	NMN
	NO	NR	P	PG	PMN
	PS	PSG	Q	QPR	R
	RI	RPR	RSI	TIMPCT	TRTM1
	TRTM2	XG	XI		

COMMON USED:	ABW	ALFA	ALNGTH	DATEJ	DELT
	IPRINT	ISPH	ISP2	ISTMC	IUTC
	MNCN	NDIM1	NDIM2	NDIM3	NTMC
	NTP	QT	RPS	RTP	SMN
	SPHERE	TEVN	TG	TM	TRTMB
	T6	T7	U0	V0	

PRØBE Analysis

Subroutine PRØBE controls the execution of both main probe and miniprobe release events. When a probe release event occurs, PRØBE saves all states, covariance matrices, etc relating to the bus and initializes all states, covariance matrices, etc for the probe under consideration. The probe state at release is then propagated forward to entry, along with the probe control covariance matrix partitions to obtain the probe state and control dispersions at entry. Next, the probe is tracked from release to entry and probe knowledge covariance matrix partitions are propagated and updated accordingly.

Let t_j be the time of probe release and let \bar{X}_j denote the nominal bus state at release. Denote all main probe quantities with a superscript zero, and all miniprobe quantities with a superscript i (for the i th miniprobe where $i = 1, 2, 3$). Then, following release, the probe states are given by

$$\bar{X}_j^0 = \bar{X}_j \quad (1)$$

$$\bar{X}_j^i = \bar{X}_j + \begin{bmatrix} 0 \\ \Delta V_j^{(i)} \end{bmatrix} \quad (2)$$

where $\Delta V_j^{(i)}$ is the velocity increment imparted to the i th miniprobe by the spin release at t_j . The velocity increment $\Delta V_j^{(i)}$ and the miniprobe release controls are used in subroutine MINIQ to compute the execution error covariance matrix $\tilde{Q}_j^{(i)}$ associated with the spin release of the i th miniprobe. Denote the bus position/velocity knowledge and control covariance matrices at release by $P_{k_j}^0$ and $P_{c_j}^0$, respectively. Then, immediately following release, the probe position/velocity knowledge and control covariance matrices are given by

$$P_{k_j}^0 = P_{k_j} \quad (3)$$

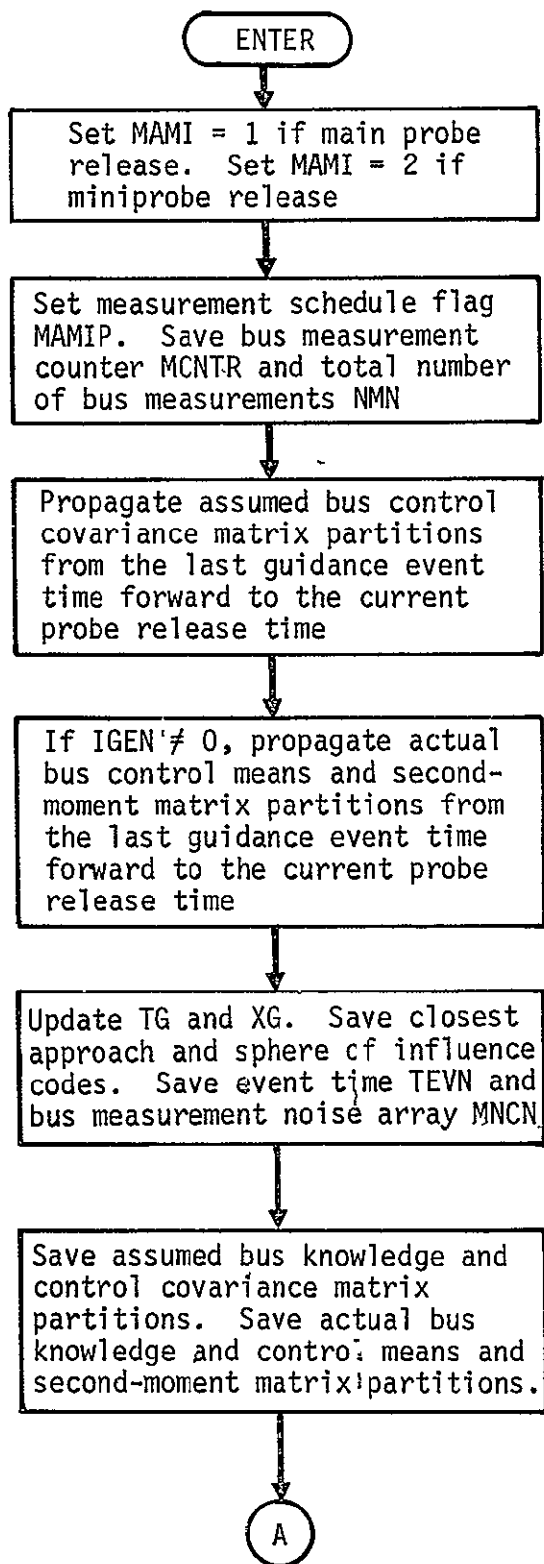
$$P_{c_j}^0 = P_{c_j} \quad (4)$$

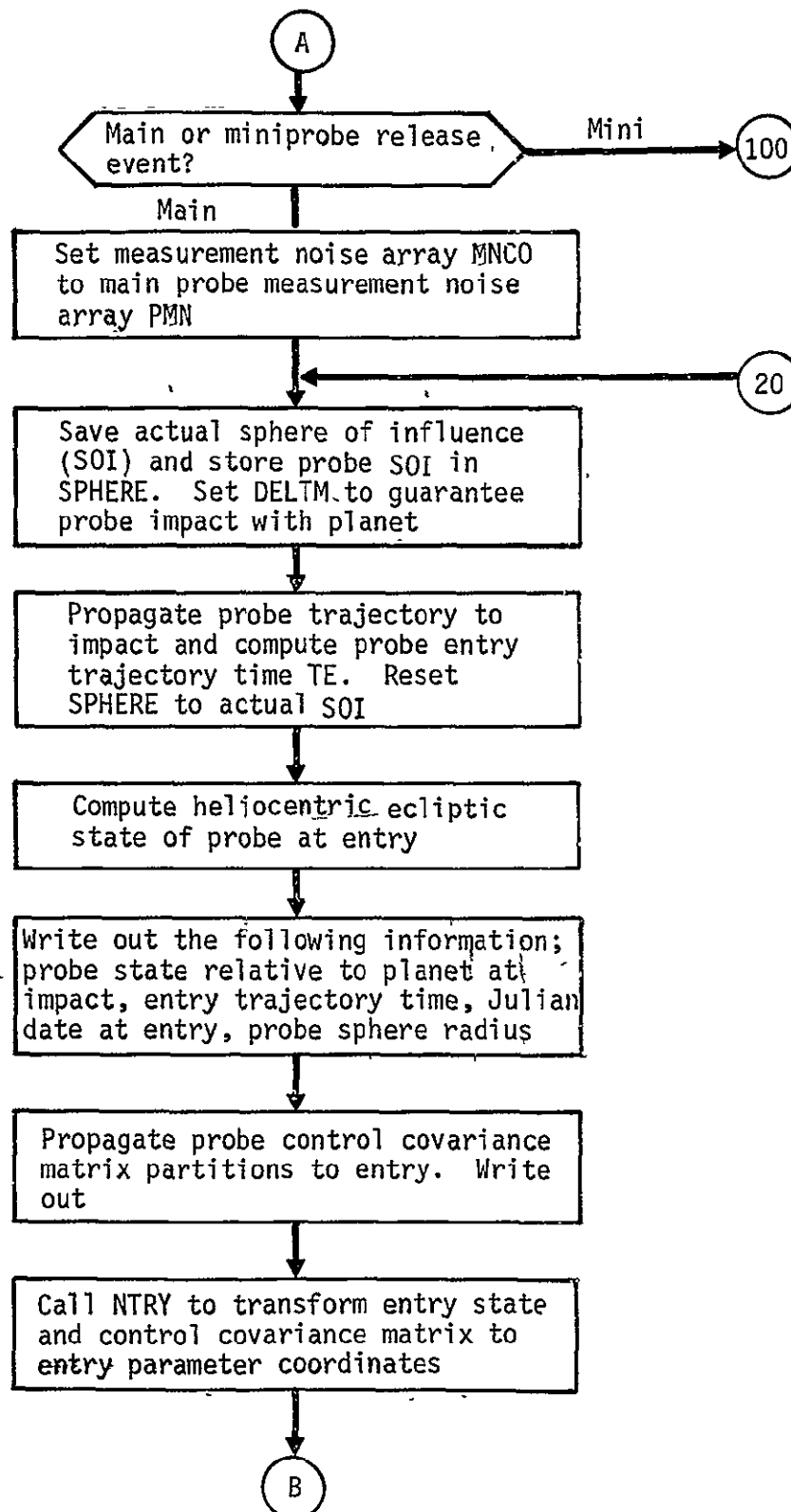
$$P_{k_j}^{(i)} = P_{k_j} + \begin{bmatrix} 0 & 0 \\ 0 & \tilde{Q}_j^{(i)} \end{bmatrix} \quad (5)$$

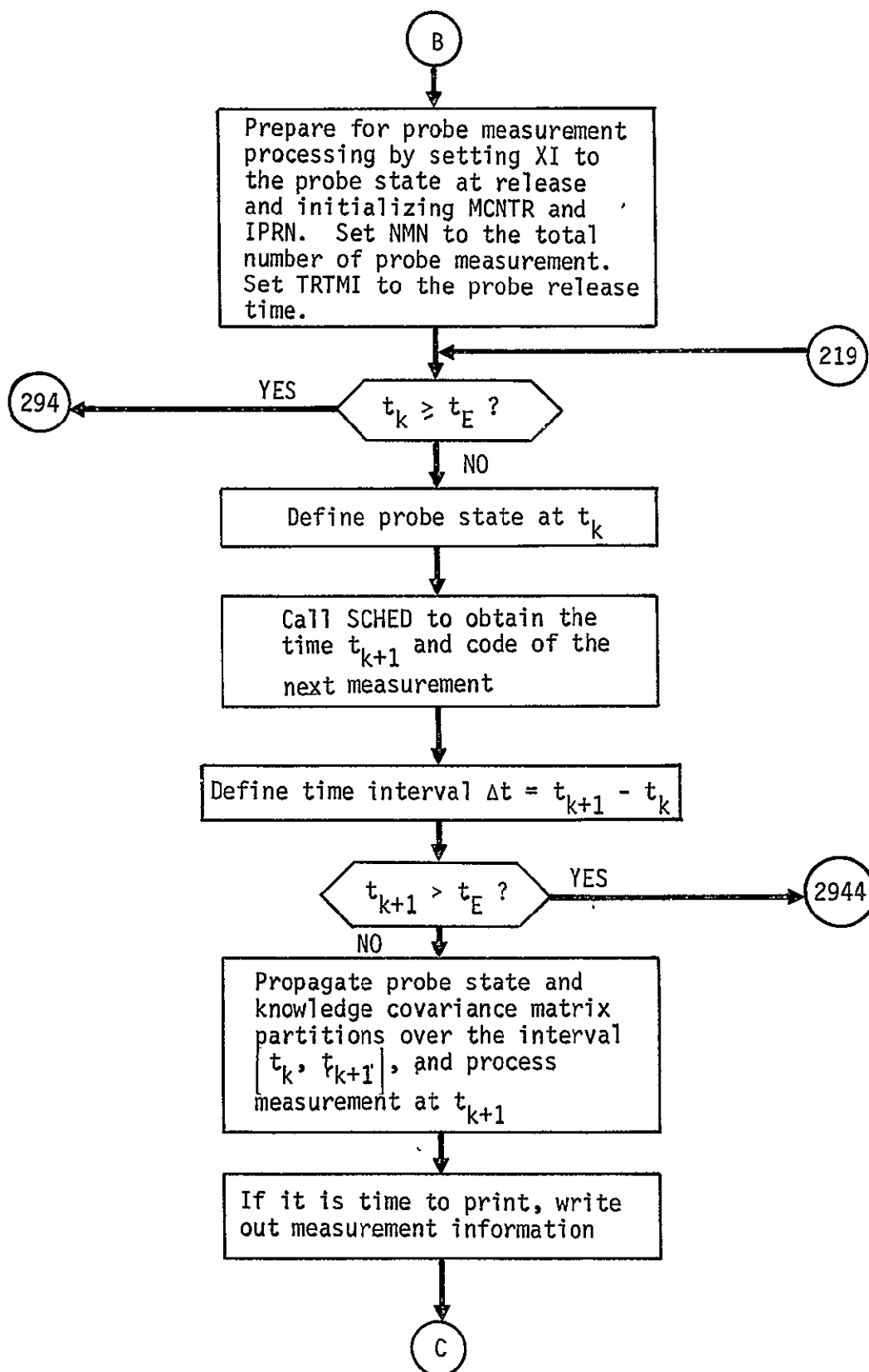
$$P_{c_j}^{(1)} = P_{k_j}^{(1)} . \quad (6)$$

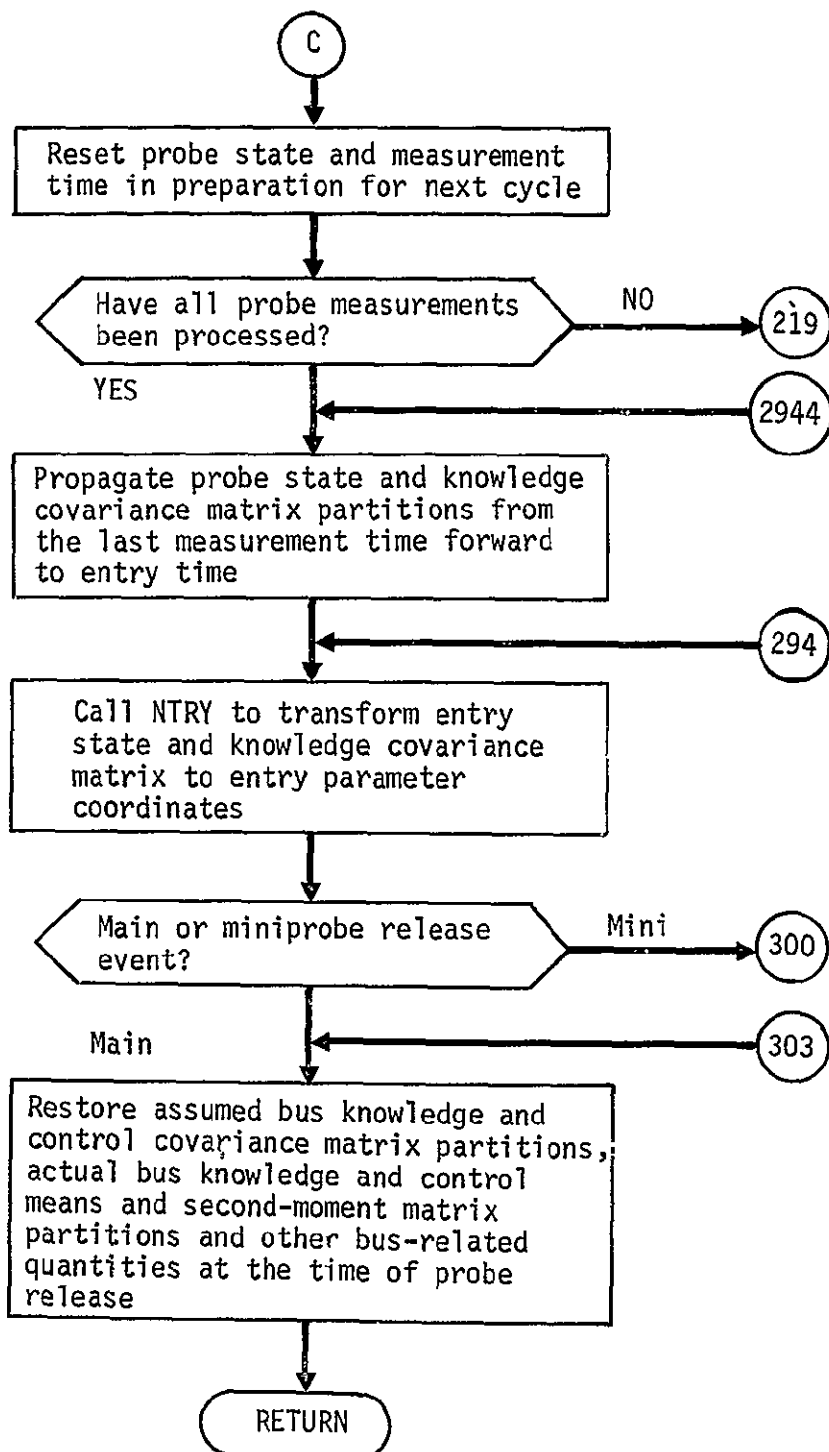
The above probe control matrices, along with all related partitions, are propagated forward to entry time t_E using the propagation equations appearing in subroutine GNAVM to obtain the probe entry dispersions. Beginning with the above knowledge covariance matrix of the probe under consideration and all related partitions, the probe knowledge covariance matrix partitions are propagated and updated as each probe measurement is processed using the update equations appearing in subrouòtine GNAVM.

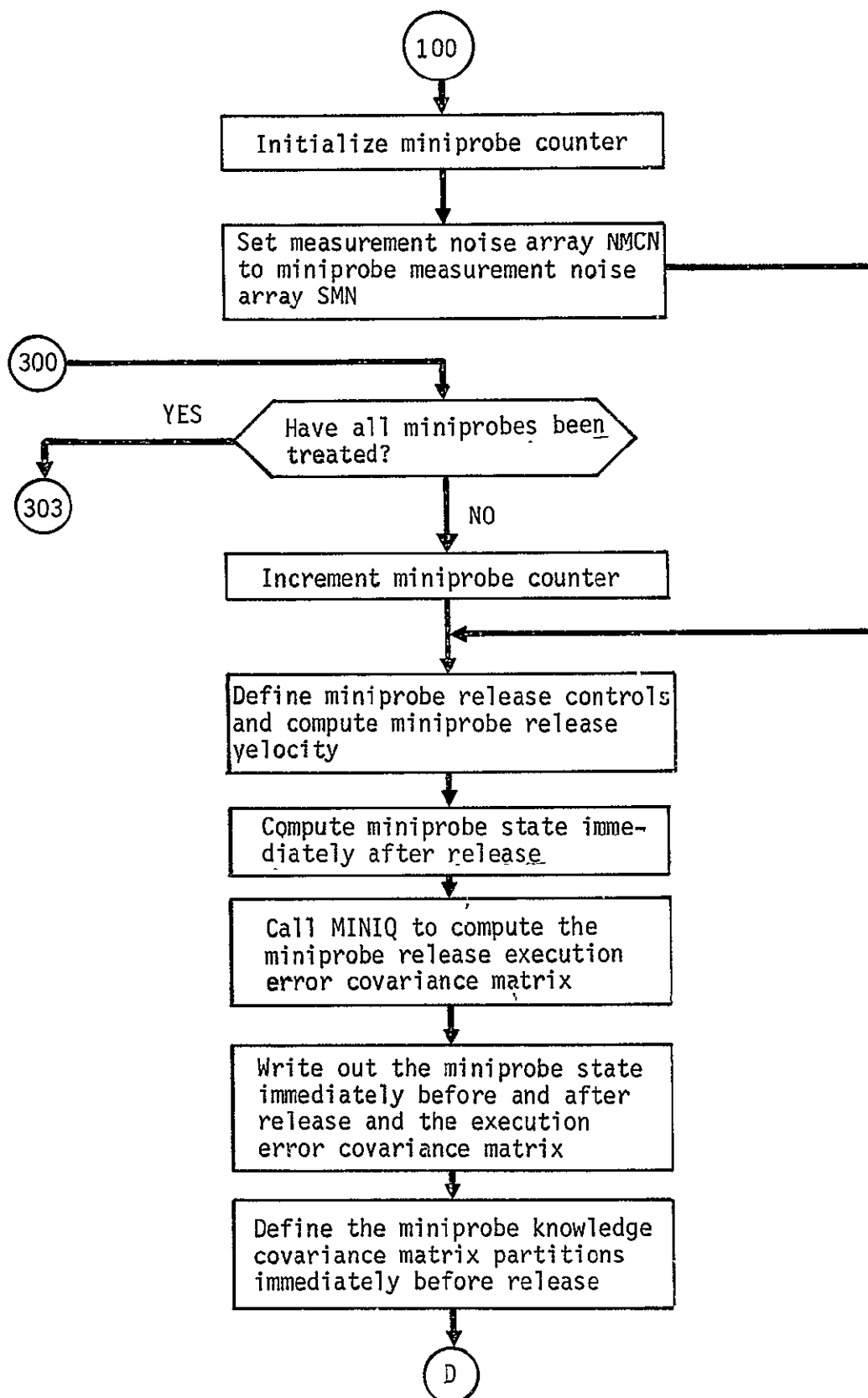
PRØBE Flow Chart

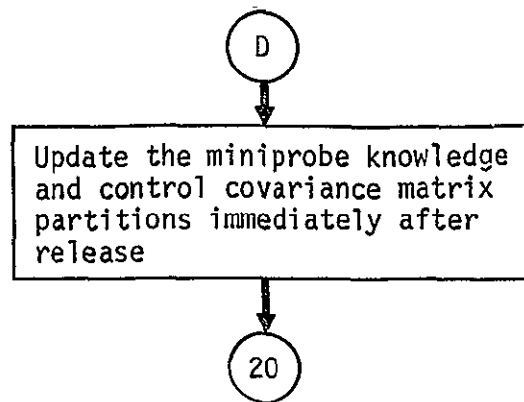












SUBROUTINE PROBES

PURPOSE: TO CONTROL THE EXECUTION OF ALL PROBE RELEASE EVENTS IN
SIMUL

CALLING SEQUENCE: CALL PROBES

SUBROUTINES SUPPORTED: SIMULL

SUBROUTINES REQUIRED: NTMS PSIM DYNOS NAVM MEAN
ORB EPHEM STMPR CORREL NTRY
SCHED TRAKS MENOS PRINT4 TPRTG
MINIQ

LOCAL SYMBOLS: AI MINI-PROBE ROLL RELEASE ANGLE
BVAL BIAS VALUE VECTOR
COSA COS(ALFA)
COSAI COS(AI)
COSD COS(DELT)
DTSUNS TEMPORARY STORAGE FOR DTSUN
DUM INTERMEDIATE VECTOR
DUMM INTERMEDIATE VARIABLE
ICL2S TEMPORARY STORAGE FOR ICL2
IPRN PRINT COUNTER FOR MEASUREMENT PROCESSING
IQL QUASI-LINEAR EVENT COUNTER
ISP2S TEMPORARY STORAGE FOR ISP2
MAMI =1 IF MAIN PROBE BEING PROCESSED
=2 IF MINI-PROBE BEING PROCESSED
MAMIP =MAMI + 1 (INPUT TO SCHED, POINTS TO PROPER
SCHEDULE OF MEASUREMENTS)
MCNTRS TEMPORARY STORAGE FOR MCNTR
MMCODE MEASUREMENT CODE
NEVENT EVENT COUNTER
NMNS TEMPORARY STORAGE FOR NMN
NMP NUMBER OF MINI-PROBE BEING PROCESSED

OLDX TEMPORARY STORAGE FOR XF
 PI MATHEMATICAL CONSTANT
 PVS PLANETOCENTRIC STATE AT PROBE SPHERE
 SADVX TEMPORARY STORAGE FOR ADEVX
 SEDVX TEMPORARY STORAGE FOR EDEVX
 SAVRM TEMPORARY STORAGE FOR AVARM
 SINA SIN(ALFA)
 SINAI SIN(AI)
 SIND SIN(DELT)
 SMNCN TEMPORARY STORAGE FOR MNCN ARRAY
 SPHERS TEMPORARY STORAGE FOR SPHERE (NTP)
 SXF1 TEMPORARY STORAGE FOR XF1
 SZF TEMPORARY STORAGE FOR ZF
 TE TIME AT THE PROBE SPHERE ____
 TEVNS TEMPORARY STORAGE FOR TEVN
 TRTM2 TIME OF NEXT MEASUREMENT OR EVENT
 VT TANGENTIAL VELOCITY OF MINI-PROBE
 XFP HELIOCENTRIC STATE AT PROBE SPHERE
 XPS PLANETOCENTRIC STATE AT PROBE SPHERE
 XSAVE HELIOCENTRIC STATE AT BEGINNING OF PROBE
 EVENT

COMMON COMPUTED/USED:

CXSU	CXSUG	CXSV	CXSVG	CXU
CXUG	CXV	CXVG	CXXS	CXXSG
DELT	ICL2	MCNTR		NMN
NO	NR	P	PG	PMN
PS	PSG	Q	R	
RI		RSI	TIMPCT	TRTM1
TRTM2	XG	XI		

COMMON USED:

ABW	ALFA	ALNGTH	DATEJ	DELT
IPRINT	ISPH	ISP2	ISTMC	IUTC
MNCN	NDIM1	NDIM2	NDIM3	NTMC

NTP	QT	RPS	RTP	SMN
SPHERE	TEVN	TG	TM	TRTMB
T6	T7	U0	V0	

PRØBS Analysis

Subroutine PRØBS controls the execution of both main probe and miniprobe release events in the simulation program SIMUL. When a probe release event occurs, PRØBS saves all nominal and actual states, deviation estimates, covariance matrices, etc relating to the bus and initializes all nominal and actual states, deviation estimates, covariance matrices, etc for the probe under consideration. The probe state at release is the propagated forward to entry, along with the probe control covariance matrix partitions, to obtain the probe state and control dispersions at entry. Next the probe is tracked from release to entry and probe knowledge covariance matrix partitions and deviation estimates are propagated and updated accordingly.

Let t_j denote the time of probe release. Denote the bus targeted nominal and most recent nominal states at release by \bar{X}_j and \tilde{X}_j , respectively. Denote actual and estimated deviations of the bus state from the most recent nominal by $\delta\tilde{X}_j$ and $\delta\hat{X}_j$, respectively.

All main probe quantities will be denoted with a superscript zero and all miniprobe quantities with a superscript i (for the i th miniprobe, where $i = 1, 2, 3$). Then following release, the probe nominal states are given by

$$\bar{X}_j^0 = \bar{X}_j \quad (1)$$

$$\tilde{X}_j^0 = \tilde{X}_j \quad (2)$$

$$\bar{X}_j^1 = \tilde{X}_j + \delta\hat{X}_j + \left[-\frac{0}{\Delta V_j^1} \right] \quad (3)$$

$$\tilde{X}_j^1 = \tilde{X}_j \quad (4)$$

where ΔV_j^1 is the velocity increment imparted to the i th miniprobe by the spin-release at t_j . The velocity increment ΔV_j^1 and the miniprobe release controls are used in subroutine MINIQ to compute the execution error covariance matrix \tilde{Q}_j^1 and the actual execution $\delta\Delta V_j^1$.

The actual and estimated deviations of the probe states from the most recent nominals are given by

$$\delta \tilde{X}_j^0 = \delta \tilde{X}_j \quad (5)$$

$$\delta \tilde{X}_j^0 = \delta \tilde{X}_j \quad (6)$$

$$\delta \tilde{X}_j^1 = 0 \quad (7)$$

$$\delta \tilde{X}_j^1 = \delta \tilde{X}_j - \delta \tilde{X}_j + \begin{bmatrix} 0 \\ \delta \Delta V_j^1 \end{bmatrix} \quad (8)$$

Denote the bus position/velocity knowledge and control covariance matrices at release by P_{K_j} and P_{c_j} , respectively. Then, immediately following release, the probe position/velocity knowledge and control covariance matrices are given by

$$P_{K_j}^0 = P_{K_j} \quad (9)$$

$$P_{c_j}^0 = P_{c_j} \quad (10)$$

$$P_{K_j}^1 = P_{K_j} + \begin{bmatrix} 0 & | & 0 \\ 0 & | & \tilde{Q}_j^1 \end{bmatrix} \quad (11)$$

$$P_{c_j}^1 = P_{K_j}^1 \quad (12)$$

The above probe control covariance matrices, along with all related partitions, are propagated forward to entry time t_E using the propagation equations appearing in subroutine NAVM to obtain the probe entry dispersions. Beginning with the above knowledge covariance matrix of the probe under consideration and all related partitions, the probe knowledge covariance matrix partitions are

propagated and updated as each probe measurement is processed using the equations appearing in subroutine NAVM. The probe estimates are propagated and updated using the equations appearing in subroutine SIMUL.

A flow chart for subroutine PRØBS is not presented since it would be quite similar to the flow chart for subroutine PRØBE (see subroutine PRØBE for details).

SUBROUTINE PSIM

PURPOSE: TO COMPUTE THE STATE TRANSITION MATRIX PARTITIONS PHI, TXXS, TXW, AND TXU OVER AN ARBITRARY INTERVAL OF TIME (TK, TK+1).

CALLING SEQUENCE: CALL PSIM(RI, RF, ISC)

ARGUMENT: ISC I CODE SPECIFYING WHICH TECHNIQUE IS TO BE USED TO COMPUTE THE STATE TRANSITION MATRIX PARTITION PHI

RF I POSITION AND VELOCITY OF THE VEHICLE AT THE END OF THE TIME INTERVAL

RI I POSITION AND VELOCITY OF THE VEHICLE AT THE BEGINNING OF THE TIME INTERVAL

SUBROUTINES SUPPORTED: SIMULL SETEVS BIAIM GUISIM GUISS
PRESIM ERRANN SETEVN GUIDM GUID
PRED PROBE PRCBES

SUBROUTINES REQUIRED: CASCAD CONC2 EPHEM MUND NDTM
ORB PCTM PLND

LOCAL SYMBOLS: D INTERMEDIATE JULIAN DATE

DELT TIME INTERVAL IN CORRECT UNITS

DUM TEMPORARY STORAGE FOR STATE TRANSITION MATRIX

IAN5 VARIABLE USED IN EXAMINING IAUG, IAUGIN

ICLC TEMPORARY STORAGE FOR ICL

ISPHC TEMPORARY STORAGE FOR ISPH

IPX TEMPORARY STORAGE FOR IPRINT

POSS DISTANCE OF THE VEHICLE FROM THE TARGET PLANET AT INITIAL TIME

RP STATE VECTOR

RS POSITION OF VEHICLE RELATIVE TO GOVERNING BODY AT INITIAL TIME

SAVE TEMPORARY STORAGE FOR ACC

THSP CONSTANT EQUAL TO SIX TIMES THE SPHERE OF INFLUENCE OF TARGET PLANET

VEC POSITION AND VELOCITY OF VEHICLE RELATIVE TO
 TARGET PLANET AT INITIAL TIME

VS VELOCITY OF VEHICLE RELATIVE TO GOVERNING
 BODY AT INITIAL TIME

COMMON COMPUTED/USED:

NO XP

COMMON COMPUTED:

PHI TXU TXXS TXW

COMMON USED:

ALNGTH	DATEJ	DELT	DTMAX	F
IAUGDC	IAUG	IBARY	ISTM1	NEOD
NB	NDIM1	NDIM2	NTP	RVS
SPHERE	TM	TRTM1	VMU	ZERO

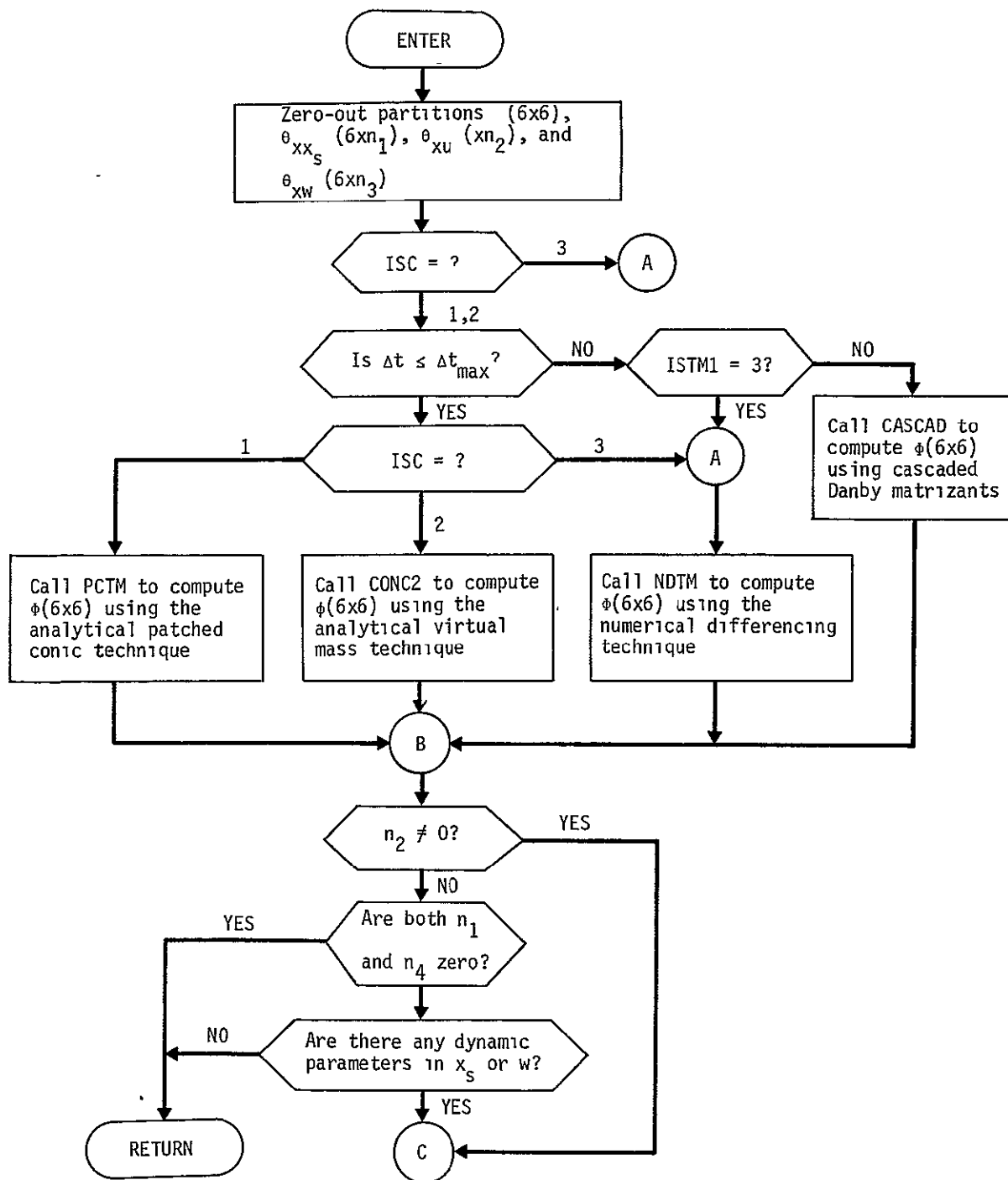
PSIM Analysis

Subroutine PSIM controls the computation of each partition appearing in the augmented state transition matrix

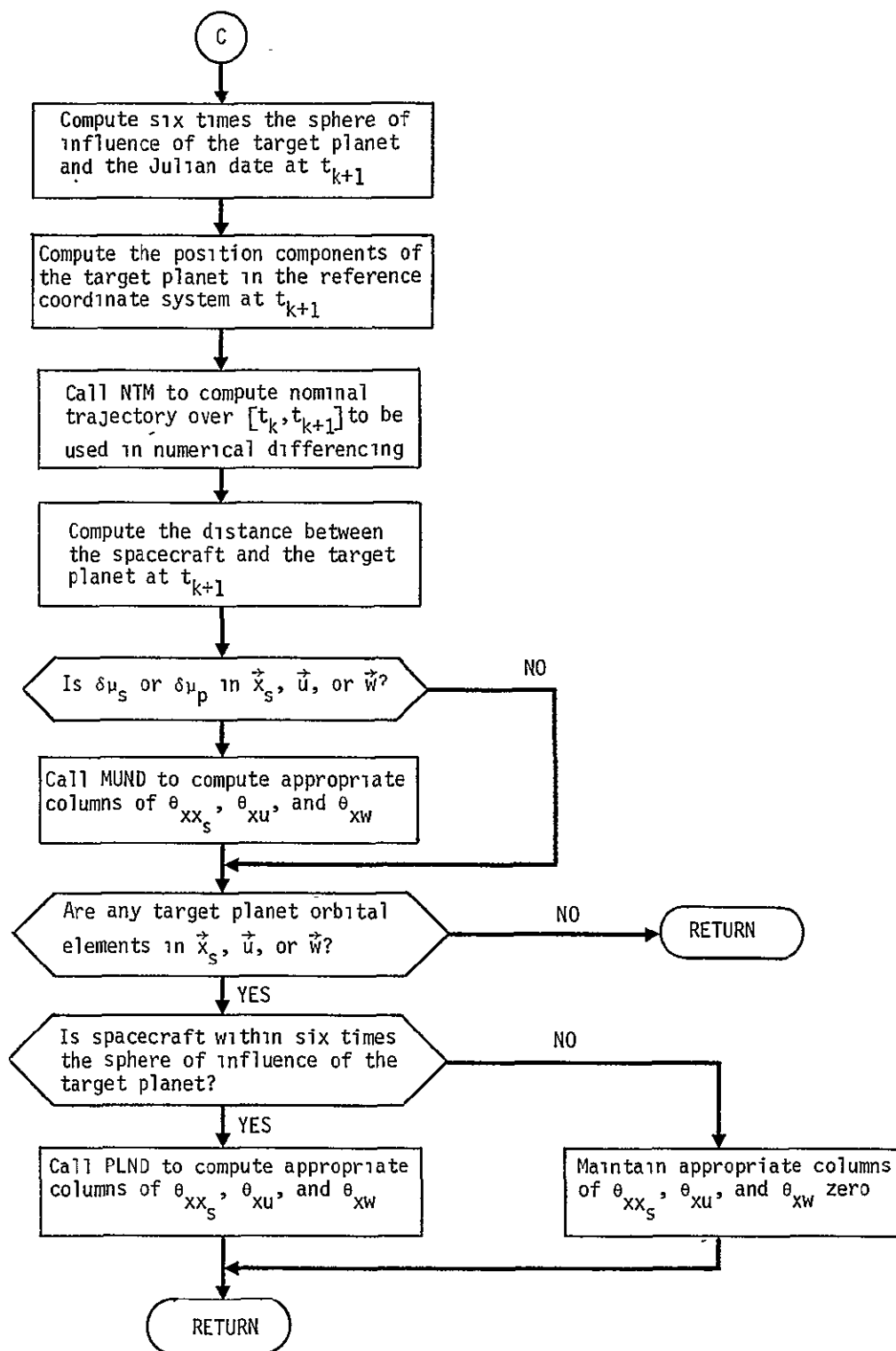
$$\Phi^A(k+1,k) = \begin{bmatrix} \Phi(k+1,k) & \theta_{xx_s}(k+1,k) & \theta_{xu}(k+1,k) & 0 & \theta_{xw}(k+1,k) \\ 0 & I & 0 & 0 & 0 \\ 0 & 0 & I & 0 & 0 \\ 0 & 0 & 0 & I & 0 \\ 0 & 0 & 0 & 0 & I \end{bmatrix}$$

The first part of the subroutine deals solely with the computation of $\Phi(k+1,k)$ by one of the three techniques -- analytical patched conic, analytical virtual mass, or numerical differencing. If an analytical technique is selected for computing $\Phi(k+1,k)$ over an interval of time greater than the maximum time interval for which the analytical technique is considered valid, we compute $\Phi(k+1,k)$ using numerical differencing or by cascading Danby matrizants.

The remaining partitions, θ_{xx_s} , θ_{xu} , and θ_{xw} , are always computed by numerical differencing. Columns in these partitions associated with target planet gravitational constant or orbital elements are computed only if the spacecraft is within six times the sphere of influence of the target planet at t_{k+1} . Otherwise, these columns are set to zero.



PSIM Flow Chart



SUBROUTINE PULCOV

PURPOSE COMPUTE EFFECTIVE EXECUTION ERROR COVARIANCE MATRIX FOR
A VELOCITY CORRECTION MODELED AS AN IMPULSE SERIES

CALLING SEQUENCE: CALL PULCOV(RIN, DELTAV, TM, QK)

ARGUMENTS: RIN(6) I INERTIAL STATE OF SPACECRAFT AT NOMINAL
TIME OF CORRECTION
DELTAV(3) I TOTAL VELOCITY INCREMENT TO BE ADDED
TM I TIME UNITS PER DAY
QK(6,6) O DEVIATION MATRIX RESULTING FROM EXECUTION
ERRORS

SUBROUTINES SUPPORTED: EXCUT EXCUTS

SUBROUTINES REQUIRED: PERHEL QCOMP

LOCAL SYMBOLS: DELR PERTURBATION IN POSITION
DELV PERTURBATION IN VELOCITY
DVFM MAGNITUDE OF FINAL PULSE
DVIM MAGNITUDE OF TYPICAL PULSE
FSER F-SERIES CONSTANT FOR PLANET
GSER G-SERIES CONSTANT FOR PLANET
HLTF STATES OF LAUNCH AND TARGET BODIES AT END
OF PROPAGATION INTERVAL
PERT CURRENT PERTURBATION
PHI STATE TRANSITION MATRIX OVER TYPICAL
INTERVAL
QQ DEVIATION MATRIX DURING PROPAGATION
THROUGH PULSES
Q TYPICAL VELOCITY EXECUTION ERROR
COVARIANCE
RF NOMINAL INERTIAL STATE OF SPACECRAFT AT
END OF TYPICAL INTERVAL
RPF PERTURBED INERTIAL STATE OF SPACECRAFT AT
END OF TYPICAL INTERVAL
R INERTIAL STATE OF SPACECRAFT AT BEGINNING

OF TYPICAL INTERVAL

T1	TIME INTERVAL BETWEEN PULSES
T2	$T1^{**2}$
T3	$T1^{**3}$
T4	$T1^{**4}$
T5	$T1^{**5}$
T6	$T1^{**6}$

COMMON USED:

DTI	DVF	DVI	FS	GG
GS	NPUL	ONE	PSIGA	PSIGB
PSIGK	PSIGS	RK	TWO	VK
ZERO				

PULCØV Analysis

PULCØV processes the control covariance through the pulsing arc to determine a measure of the probabilistic deviation of the corrected trajectory from the desired trajectory resulting from execution errors.

The pulsing arc itself is computed in PREPUL. It consists of $N_p - 1$ pulses $\overrightarrow{\Delta v}_1$ and a final pulse $\overrightarrow{\Delta v}_f$ satisfying

$$(N_p - 1) \overrightarrow{\Delta v}_1 + \overrightarrow{\Delta v}_f = \overrightarrow{\Delta v} \quad (1)$$

where $\overrightarrow{\Delta v}$ is the equivalent single impulse. The pulses are separated by a time interval Δt_1 . The duration of the entire sequence of pulses is given by $\Delta T = (N_p - 1) \Delta t_1$.

PULCØV must compute the execution error matrices Q , Q_f corresponding to the nominal pulse $\overrightarrow{\Delta v}_1$ and the final pulse $\overrightarrow{\Delta v}_f$ respectively. The error model for the engine is defined by the input specifications

$$\begin{aligned} \sigma_k^2 &= \text{proportionality error} \\ \sigma_k^2 &= \text{resolution error} \\ \sigma_\alpha^2 &= \text{first pointing error} \\ \sigma_\beta^2 &= \text{second pointing error} \end{aligned}$$

The execution error matrix measuring the probabilistic deviation of the actual velocity increment from the desired velocity increment is computed by QCØMP.

The exact equations defining the propagation of the covariance matrix are recursive in nature. If P_k^+ is the control covariance immediately after the k^{th} pulse, the covariance will propagate to the time of the next pulse t_{k+1} by the formula

$$P_{k+1}^- = \Phi_{k+1,k} P_k^+ \Phi_{k+1,k}^T \quad (2)$$

where $\Phi_{k+1,k}$ is the 6x6 state transition matrix relating perturbations at t_{k+1} to perturbations at t_k . Adding the pulse at t_{k+1} expands the covariance by

$$P_{k+1}^+ = P_{k+1}^- + \begin{bmatrix} 0 & 1 & 0 \\ - & - & - \\ 0 & 1 & Q \end{bmatrix} \quad (3)$$

where Q is set equal either to the nominal or final form of Q .

To start the process the control covariance following the first pulse is given by

$$P_1^+ = \begin{bmatrix} 0 & 1 & 0 \\ - & - & - \\ 0 & 1 & Q \end{bmatrix} \quad (4)$$

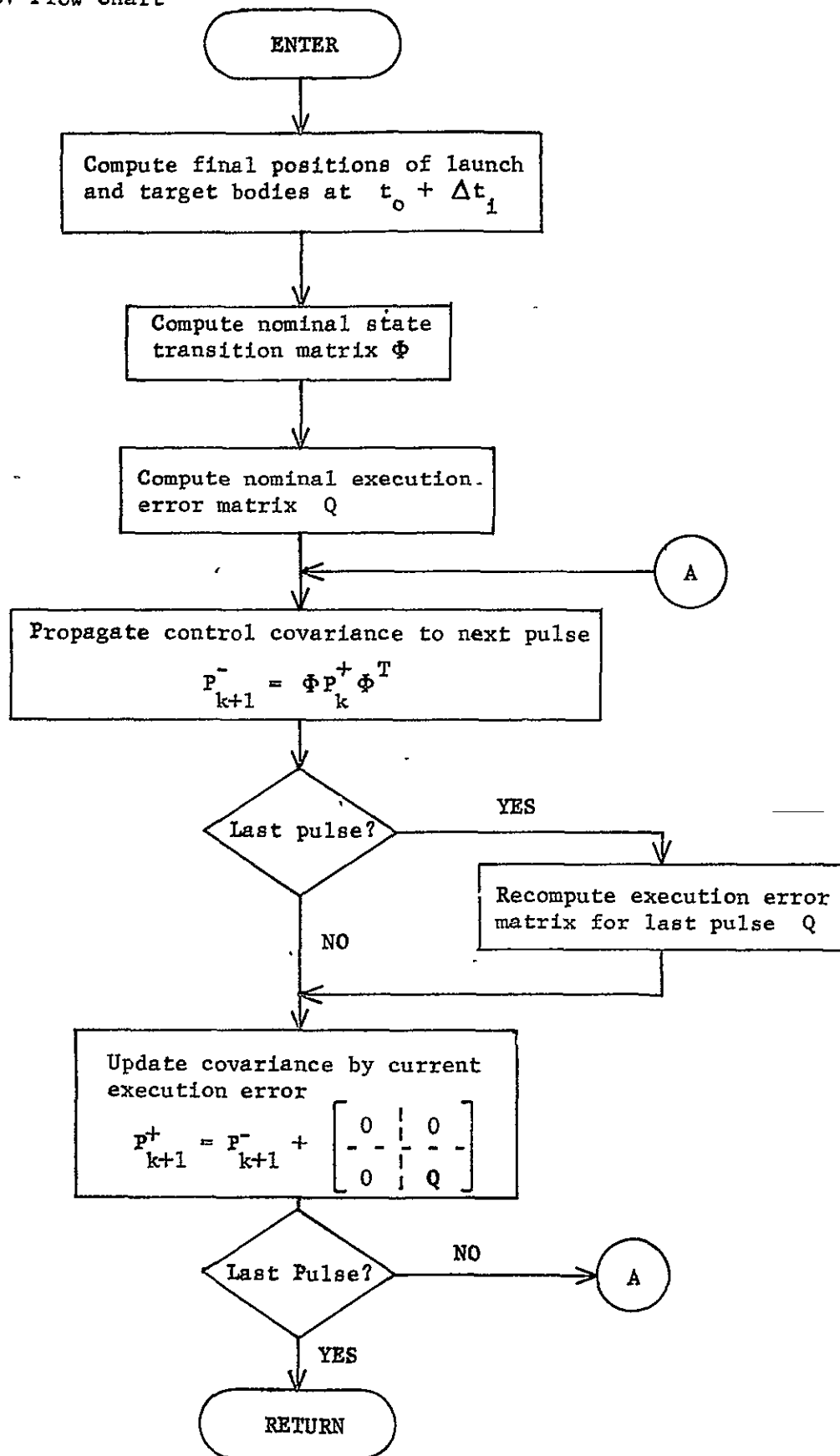
For efficiency one simplification is made in the process. Instead of recomputing the state transition matrix over each interval, the value of that matrix is held constant at the value corresponding to the "average interval". To explain this, let the state of the spacecraft at the time t_0 of the impulsive $\Delta \vec{v}$ computation be denoted \vec{r}_0, \vec{v}_0 . Then the "average interval" is defined to be the perturbed heliocentric trajectory (PERHEL) resulting from the propagation of the state $(\vec{r}_0, \vec{v}_0 + \frac{1}{2} \Delta \vec{v})$ over the interval $(t_0, t_0 + \Delta t_1)$.

The constant state transition matrix Φ is computed by numerical differencing. The initial state $(\vec{r}_0, \vec{v}_0 + \frac{1}{2} \Delta \vec{v})$ is first propagated over the Δt_1 time interval (using PERHEL) resulting in the state (\vec{r}_f, \vec{v}_f) . Then the x-component of initial position is perturbed by Δx , leading to a final state of $(\vec{r}_{pf}, \vec{v}_{pf})$ upon propagation. The first column of the matrix is then computed by

$$\Phi_1 = \begin{bmatrix} \frac{\vec{r}_{pf} - \vec{r}_f}{\Delta x} & , & \frac{\vec{v}_{pf} - \vec{v}_f}{\Delta x} \end{bmatrix}^T \quad (5)$$

The other columns of Φ are computed by similar computations using the remaining components of position and velocity $(y, z, \dot{x}, \dot{y}, \dot{z})$.

PULCOV Flow Chart



SUBROUTINE PULSEX

PURPOSE: TO CONTROL EXECUTION OF THE PULSING ARC MODEL.

CALLING SEQUENCE: CALL PULSEX(RIN,DELTAV,RE,TM,IRE)

ARGUMENTS: RIN(6) I INERTIAL STATE OF SPACECRAFT AT NOMINAL
TIME OF CORRECTION

DELTAV(3) I TOTAL VELOCITY INCREMENT TO BE ADDED

RE(6) O FINAL INERTIAL STATE OF SPACECRAFT (IRE)

TM I TIME UNITS PER DAY

IRE I FLAG DETERMINING FINAL STATE
=0 RETURN FINAL STATE AT END OF PULSE ARC
=1 RETURN FINAL STATE AT ARC MIDPOINT

SUBROUTINES SUPPORTED: EXCUTE EXCUTS EXCUT

SUBROUTINES REQUIRED: QAREL PERHEL

LOCAL SYMBOLS: A SEMIMAJOR AXIS

DTS TIME INTERVAL IN TIME UNITS

DT DUMMY VARIABLE FOR OUTPUT

E ECCENTRICITY

FSER F-SERIES CONSTANT

GSER G-SERIES CONSTANT

HLTF STATES OF LAUNCH AND TARGET BODIES AT END
OF PROPAGATION INTERVAL

HLTI STATES OF LAUNCH AND TARGET BODIES AT
BEGINNING OF PROPAGATION INTERVAL

IPUL PULSE COUNTER

PP UNIT VECTOR TO PERIAPSIS

QQ UNIT VECTOR IN ORBITAL PLANE NORMAL TO PP

RB INERTIAL STATE OF SPACECRAFT AT BEGINNING
OF PROPAGATION INTERVAL

TA TRUE ANOMALY

TFP TIME FROM PERIAPSIS

TK TIME FROM START OF PULSING ARC
 TS INTERMEDIATE VARIABLE
 T1 TIME INTERVAL FROM MIDPOINT OF ARC
 T2 $T1^{**2}$
 T3 $T1^{**3}$
 T4 $T1^{**4}$
 T5 $T1^{**5}$
 T6 $T1^{**6}$
 W ARGUMENT OF PERIAPSIS
 WW UNIT NORMAL TO ORBITAL PLANE
 XI INCLINATION
 XN LONGITUDE OF ASCENDING NODE

COMMON USED:

DTI	DVF	DVI	FS	GG
GS	NPUL	ONE	PULT	RK
TWO	VK	ZERO		

PULSEX Analysis

PULSEX is responsible for the actual execution of the pulsing arc. Experiments have shown that adding an impulsive $\Delta \vec{v}$ at time t_0 may be approximated quite closely by centering an equivalent sequence of smaller impulses about the nominal time t_0 .

This equivalent sequence of thrusts is computed by PREPUL. It consists of $N_p - 1$ pulses $\Delta \vec{v}_i$ and a final pulse $\Delta \vec{v}_f$ satisfying

$$(N_p - 1) \Delta \vec{v}_i + \Delta \vec{v}_f = \Delta \vec{v} \quad (1)$$

The pulses are separated by a time interval Δt_i . The duration of the entire sequence of pulses is given by $\Delta T = (N_p - 1) \Delta t_i$.

For efficiency the perturbed heliocentric conic propagator PERHEL is used to propagate the trajectory between pulses. PERHEL requires the positions of the launch and target bodies at the beginning and end of each propagation interval. PREPUL stores the position and velocity of the launch and target bodies at the reference time t_0 : $(\vec{r}_{LO}, \vec{v}_{LO})$ and $(\vec{r}_{TO}, \vec{v}_{TO})$ and stores the constants of the f and g series for those states $(f_{Lk}, g_{Lk}, f_{Tk}, g_{Tk}, k=1,6)$. The position of the launch body at some time t relative to the reference time t_0 is then given by

$$\vec{r}_L(t) = f_L(t) \vec{r}_{LO} + g_L(t) \vec{v}_{LO} \quad (2)$$

$$\text{where } f_L(t) = \sum_{k=0}^6 f_{Lk} t^k \quad (3)$$

$$g_L(t) = \sum_{k=1}^6 g_{Lk} t^k$$

with similar equations holding for the target body.

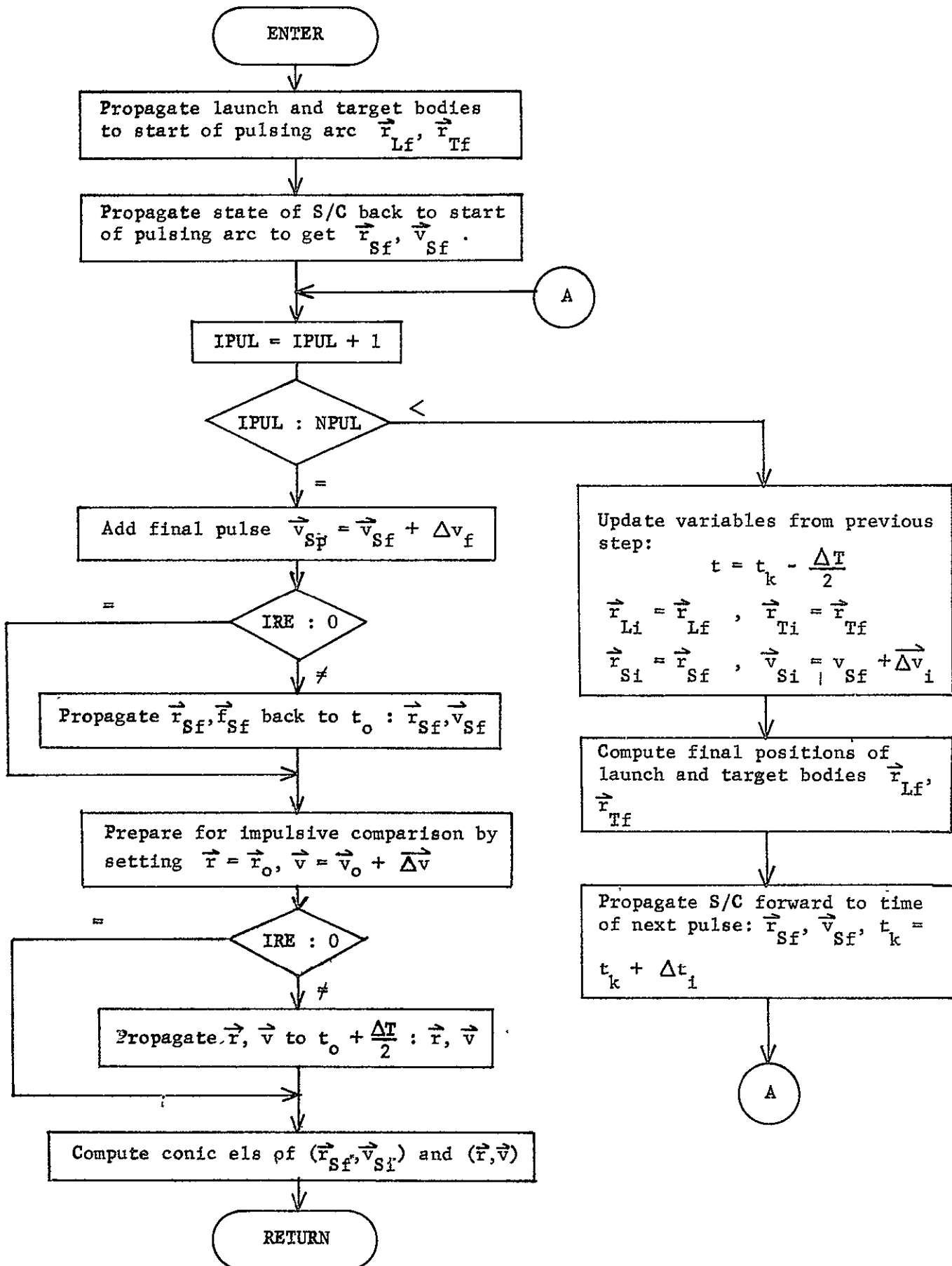
The procedure of PULSEX is straightforward. The positions of the launch and target bodies are computed at the time the pulsing arc should begin: $t_B = t_0 - \Delta T/2$. PERHEL is then called to propagate the spacecraft from t_0 backwards to t_B . The actual pulsing arc cycle is now entered. The nominal velocity increment $\Delta \vec{v}_i$ is added to the current velocity impulsively

$$\vec{v} = \vec{v} + \Delta \vec{v}_i \quad (4)$$

and the resulting state (\vec{r}, \vec{v}) is propagated forward over the time interval Δt_1 by PERHEL. Another pulse is added and the process repeated until $N_p - 1$ pulses have been added. Finally a pulse of $\vec{\Delta v}_f$ is added.

Two options are now permitted. If $IRE = 0$, the final state is not altered (NOMNAL). If $IRE = 1$, the final state is propagated backwards back to t_0 for use in ERRAN and SIMUL.

Finally CAREL is called to compute the conic elements of the final state. For comparison purposes, the impulsive $\vec{\Delta v}$ is added to the state at t_0 , propagated to the final time $t_E = t_0 + \Delta T/2$ by PERHEL, and those elements computed.



SUBROUTINE QCOMP

PURPOSE: TO COMPUTE THE EXECUTION ERROR COVARIANCE MATRIX FOR A
VELOCITY CORRECTION.

CALLING SEQUENCE: CALL QCOMP(V,EM,Q)

ARGUMENT: V I VELOCITY CORRECTION
Q O EXECUTION ERROR MATRIX
EM I ERROR MODEL (SIGRES,SIGPRO,SIGALP,SIGBET)

SUBROUTINES SUPPORTED: BIAIM GUISIM PULCOV GUIDM

LOCAL SYMBOLS: AU SIGALP/U2
BRK SIGPRO+ SIGRES/R2
BU SIGBET/U2
R2 U2+Z2
U2 X2+Y2
X2 V(1) SQUARED
Y2 V(2) SQUARED
Z2 V(3) SQUARED

QCOMP Analysis

Subroutine QCOMP computes the execution error covariance matrix \tilde{Q}_j for a velocity correction $\vec{\Delta V} = (\Delta V_x, \Delta V_y, \Delta V_z)$ occurring at time t_j . If the execution error is assumed to have form

$$\delta \vec{\Delta V} = k \vec{\Delta V} + s \frac{\vec{\Delta V}}{\Delta V} + \delta \vec{\Delta V}_{\text{pointing}}$$

where k is the proportionality error and s is the resolution error, then the elements of the \tilde{Q}_j matrix are given by

$$\tilde{Q}_{11} = \Delta V_x^2 \left[\sigma_k^2 + \frac{\sigma_s^2}{\rho^2} \right] + \frac{\Delta V_y^2 \rho^2 \sigma_{\delta\alpha}^2}{\mu^2} + \frac{\Delta V_x^2 \Delta V_z^2 \sigma_{\delta\beta}^2}{\mu^2}$$

$$\tilde{Q}_{12} = \tilde{Q}_{21} = \Delta V_x \Delta V_y \left[\sigma_k^2 + \frac{\sigma_s^2}{\rho^2} - \frac{\rho^2 \sigma_{\delta\alpha}^2}{\mu^2} + \frac{\Delta V_z^2 \sigma_{\delta\beta}^2}{\mu^2} \right]$$

$$\tilde{Q}_{13} = \tilde{Q}_{31} = \Delta V_x \Delta V_z \left[\sigma_k^2 + \frac{\sigma_s^2}{\rho^2} - \sigma_{\delta\beta}^2 \right]$$

$$\tilde{Q}_{22} = \Delta V_y^2 \left[\sigma_k^2 + \frac{\sigma_s^2}{\rho^2} \right] + \frac{\Delta V_x^2 \rho^2 \sigma_{\delta\alpha}^2}{\mu^2} + \frac{\Delta V_y^2 \Delta V_z^2 \sigma_{\delta\beta}^2}{\mu^2}$$

$$\tilde{Q}_{23} = \tilde{Q}_{32} = \Delta V_y \Delta V_z \left[\sigma_k^2 + \frac{\sigma_s^2}{\rho^2} - \sigma_{\delta\beta}^2 \right]$$

$$\tilde{Q}_{33} = \Delta V_z^2 \left[\sigma_k^2 + \frac{\sigma_s^2}{\rho^2} \right] + \mu^2 \sigma_{\delta\beta}^2$$

where $\mu^2 = \Delta V_x^2 + \Delta V_y^2$, $\rho^2 = \mu^2 + \Delta V_z^2$, and σ_s^2 , σ_k^2 , $\sigma_{\delta\alpha}^2$, and $\sigma_{\delta\beta}^2$ are the variances associated with the resolution, proportionality, and two pointing errors, respectively.

PROGRAM QUASI

PURPOSE: PERFORM QUASI-LINEAR FILTERING EVENT IN THE SIMULATION
 PROGRAM

CALLING SEQUENCE: CALL QUASI

SUBROUTINES SUPPORTED: SIMULL

LOCAL SYMBOLS:

COMMON COMPUTED/USED:	ADEVX	EDEVX	NQE	XF1
-----------------------	-------	-------	-----	-----

COMMON COMPUTED:	TRTM1	XI1	XI
------------------	-------	-----	----

COMMON USED:	ADEVXS	EDEVXS	NDIM1	TEVN	W
	XF	XSL	ZERO		

QUASI Analysis

At a quasi-linear filtering event the most recent nominal trajectory is updated by using the most recent state deviation estimate. If $\tilde{\mathbf{X}}_j^-$ is the most recent nominal position/velocity state immediately preceding the event at time t_j , and if $\delta\tilde{\mathbf{X}}_j^-$ is the position/velocity deviation estimate, then immediately following the quasi-linear filtering event, the most recent nominal position/velocity state is given by

$$\tilde{\mathbf{X}}_j^+ = \tilde{\mathbf{X}}_j^- + \delta\tilde{\mathbf{X}}_j^-$$

The estimated and actual deviations from the most recent nominal trajectory must also be updated:

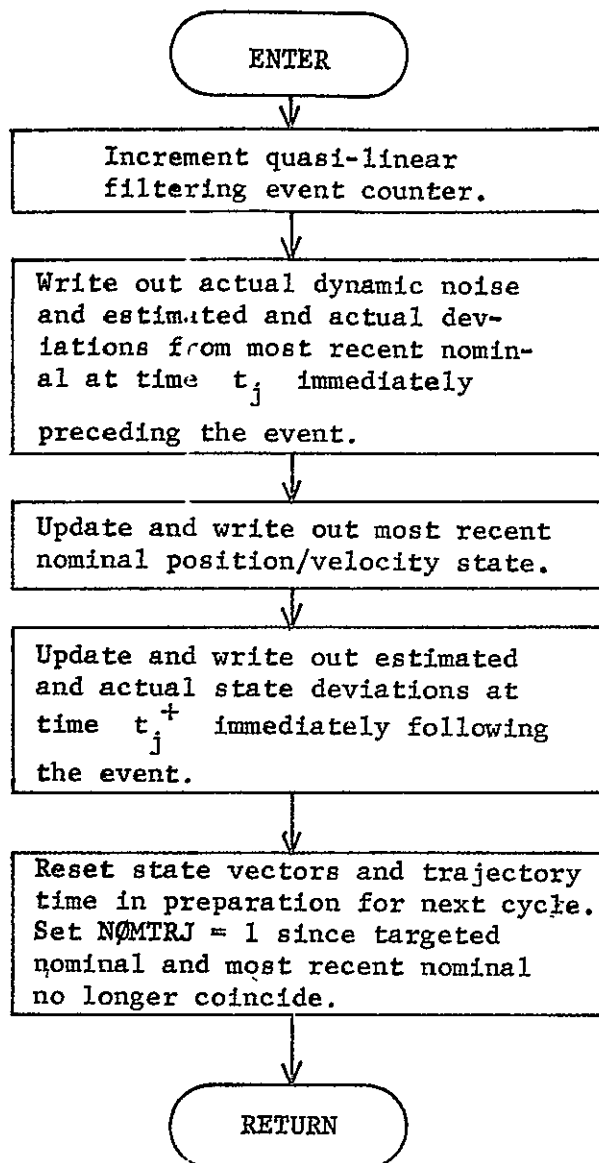
$$\delta\tilde{\mathbf{X}}_j^+ = 0$$

$$\delta\tilde{\mathbf{X}}_j^+ = \delta\tilde{\mathbf{X}}_j^- - \delta\tilde{\mathbf{X}}_j^-$$

A quasi-linear filtering event in no way alters the knowledge and control uncertainties at time t_j . Thus knowledge covariance P_{k_j} and control covariance P_{c_j} remain constant across a quasi-linear filtering event.

Furthermore, since no velocity correction is performed, the (most recent) targeted nominal $\tilde{\mathbf{X}}_j$ is unchanged. Neither is the solve-for parameter state updated at a quasi-linear filtering event.

QUASI Flow Chart



FUNCTION RNUM

PURPOSE: TO RETURN RANDOM NUMBERS ON A NORMAL DISTRIBUTION WITH MEAN ZERO AND STANDARD DEVIATION SIGMA.

CALLING SEQUENCE: Z=RNUM(SIGMA)

ARGUMENT: SIGMA I STANDARD DEVIATION

SUBROUTINES SUPPORTED: SIMULL

LOCAL SYMBOLS:	A	SUM OF TWELVE RANDOM NUMBERS BETWEEN ZERO AND ONE
	NX	CONTROLLING INTEGER
	N	INTERMEDIATE INTEGER
	Q	INTERMEDIATE VARIABLE
	RNUM	RANDOM NUMBER FROM NORMAL DISTRIBUTION WITH MEAN ZERO AND STANDARD DEVIATION SIGMA
	RR	INTERMEDIATE VARIABLE
	SS	INTERMEDIATE VARIABLE
	WW	INTERMEDIATE VARIABLE
	W1	INTERMEDIATE VARIABLE
	YY	INTERMEDIATE VARIABLE
	Y1	INTERMEDIATE VARIABLE
	ZZ	INTERMEDIATE VARIABLE
	Z1	INTERMEDIATE VARIABLE

RNUM Analysis

Function subprogram RNUM supplies random numbers on a normal distribution with near zero and standard deviation σ .

Twelve random numbers X_i between 0 and 1 are computed, which are then used to compute the returned random number RNUM using the following equation:

$$\text{RNUM} = \left[\sum_{i=1}^{12} X_i - 6 \right] \cdot \sigma$$

SUBROUTINE SAOCS

PURPOSE: TO COMPUTE SINES AND COSINES OF SPIN-AXIS RIGHT ASCENSION AND DECLINATION GIVEN SPIN-AXIS ORIENTATION MODE

ARGUMENT: CSDCSA 0 COSINE OF ECLIPTIC DECLINATION OF SPIN AXIS

CSRASA 0 COSINE OF ECLIPTIC RIGHT ASCENSION OF SPIN AXIS

DJR I JULIAN DATE EPOCH1900 OF MINIPROBE RELEASE

ISAO I SPIN AXIS ORIENTATION MODE INDICATOR

NP I NUMBER OF TARGET PLANET

SNDCSA 0 SINE OF ECLIPTIC DECLINATION OF SPIN AXIS

SNRASA 0 SINE OF ECLIPTIC RIGHT ASCENSION OF SPIN AXIS

UCNTRL I MINIPROBE RELEASE CONTROL VECTOR

VSCRPM I PLANETOCENTRIC ECLIPTIC VELOCITY VECTOR OF BUS AT MINIPROBE RELEASE IN KM/SEC

SUBROUTINES SUPPORTED: TPPROP TPRTRG

SUBROUTINES REQUIRED: EPHEM ORB SCAR

LOCAL SYMBOLS: HDOTV DOT PRODUCT OF PLANETOCENTRIC BUS VELOCITY WITH CROSS PRODUCT OF PLANETOCENTRIC VECTOR TO SUN

ONE CONSTANT 1.0

TEMP1 MAGNITUDE OF CROSS PRODUCT OF VECTOR TO SUN BY ECLIPTIC POLE VECTOR

VSCRPM MAGNITUDE OF PLANETOCENTRIC VELOCITY OF BUS AT MINIPROBE RELEASE

ZERO CONSTANT 0.

COMMON COMPUTED: NO

COMMON USED: XP

SUBROUTINE SAVMAT

PURPOSE: TO STORE N VALUES OF VECTOR A IN VECTOR B

CALLING SEQUENCE: CALL SAVMAT(A,B,N)

ARGUMENTS: A I VECTOR TO BE STORED

B O VECTOR IN WHICH A IS STORED

N I NUMBER OF ELEMENTS TO BE TRANSFERRED

SUBROUTINES SUPPORTED: PRED GENGID

SUBROUTINE SCAD

PURPOSE: TO CALCULATE BOTH SINE AND COSINE OF AN ANGLE IN DEGREES

ARGUMENT: CSA 0 COSINE OF ANGLE
DA I ANGLE IN DEGREES
SNA 0 SINE OF ANGLE

SUBROUTINES SUPPORTED: DIMPCP IMPCT

LOCAL SYMBOLS: DTR CONVERSION FACTOR FROM DEGREES TO RADIANS
ONE CONSTANT ONE
QREV 90. DEGREES
RA ANGLE IN RADIANS
REV 360. DEGREES
THQREV 270. DEGREES

SUBROUTINE SCAR

PURPOSE: TO CALCULATE BOTH SINE AND COSINE OF AN ANGLE GIVEN IN RADIANS

ARGUMENT: CSA 0 COSINE OF ANGLE

RA I ANGLE IN RADIANS

SNA 0 SINE OF ANGLE

SUBROUTINES SUPPORTED: SAOCS TPPROP

LOCAL SYMBOLS: HALFPI CONSTANT $\pi/2$

ONE CONSTANT 1.

THHFPI CONSTANT $3 \cdot \pi/2$.

TWOPI CONSTANT $2 \cdot \pi$

SUBROUTINE SCHED

PURPOSE: TO DETERMINE WHAT TYPE OF MEASUREMENT IS TO BE TAKEN
NEXT AND AT WHAT TIME IT WILL OCCUR.

CALLING SEQUENCE: CALL SCHED(T1,T2,MMCODE,IPOINT)

ARGUMENT: MCODE 0 MEASUREMENT MODEL CODE

T1 I PRESENT TRAJECTORY TIME

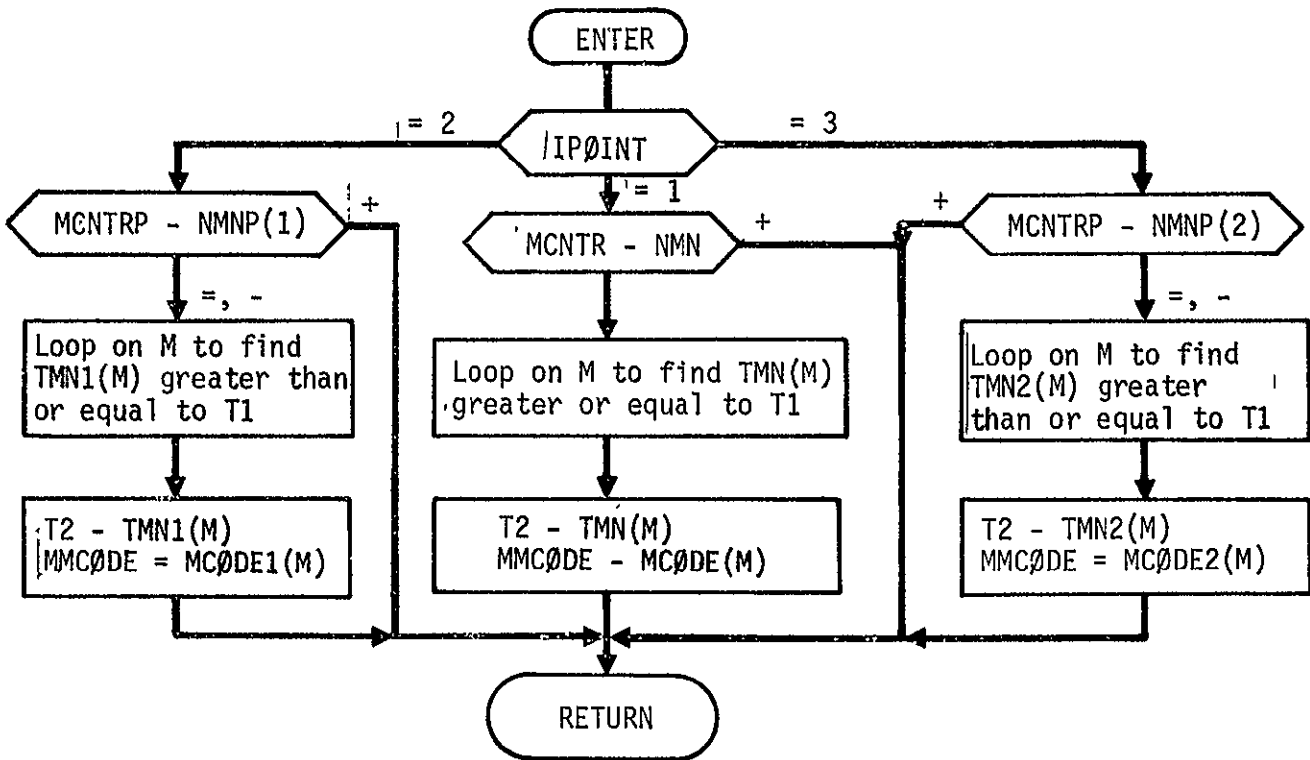
T2 0 TRAJECTORY TIME AT WHICH THE NEXT
MEASUREMENT OCCURS

IPOINT I =1 FOR BUS
 =2 FOR MAIN PROBE
 =3 FOR MINI-PROBE

SUBROUTINES SUPPORTED: SIMULL ERRANN

LOCAL SYMBOLS: M INDEX

COMMON USED:	MCNTR	MCODE	NMN	TMN	
	MCNTRP	NMNP	TMN1	MCODE1	TMN2
	MCODE2				



SUBROUTINE SERIE**PURPOSE:** TO COMPUTE THE TRANSCENDENTAL FUNCTIONS USED IN FLITE.**CALLING SEQUENCE:** CALL SERIE(X,SX,CX)

ARGUMENTS:	X	I	INDEPENDENT VARIABLE
	SX	O	BATTIN S-FUNCTION OF X
	CX	O	BATTIN C-FUNCTION OF X

SUBROUTINES SUPPORTED: FLITE**SUBROUTINES REQUIRED:** NONE

LOCAL SYMBOLS:	COSH	STATEMENT FUNCTION FOR HYPERBOLIC COSINE
	SINH	STATEMENT FUNCTION FOR HYPERBOLIC SINE
	E	SQRT OF ABS VALUE OF X

SERIE Analysis

SERIE computes the transcendental functions $S(x)$ and $C(x)$ used in the FLITE program in the solution of Lambert's theorem.

The functions $S(x)$ and $C(x)$ are defined by

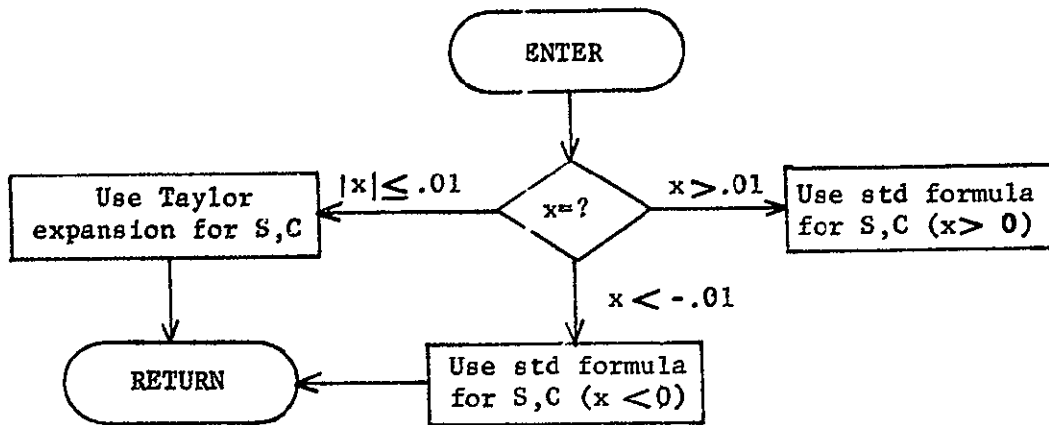
$$\begin{aligned}
 S(x) &= \frac{\sqrt{x} - \sin \sqrt{x}}{x^3} & x > 0 \\
 &= \frac{\sinh \sqrt{-x} - \sqrt{-x}}{\sqrt{-x}^3} & x < 0 \\
 &= \frac{1}{6} & x = 0
 \end{aligned} \tag{1}$$

$$\begin{aligned}
 C(x) &= \frac{1 - \cos \sqrt{x}}{x} & x > 0 \\
 &= \frac{\cosh \sqrt{-x} - 1}{-x} & x < 0 \\
 &= \frac{1}{2} & x = 0
 \end{aligned} \tag{2}$$

For small values of $|x|$ the Taylor series expansions are used

$$\begin{aligned}
 S(x) &= \frac{1}{3!} - \frac{x}{4!} + \frac{x^2}{5!} + \dots \\
 C(x) &= \frac{1}{2!} - \frac{x}{3!} + \frac{x^2}{4!} + \dots
 \end{aligned} \tag{3}$$

SERIE Flow Chart



SUBROUTINE SETEVN

PURPOSE PERFORM ALL COMPUTATIONS COMMON TO MOST EVENTS IN THE
ERROR ANALYSIS PROGRAM

CALLING SEQUENCE: CALL SETEVN(RI,TEVN,NCODE)

ARGUMENT: NCODE I EVENT CODE

RI I TARGETED NOMINAL SPACECRAFT STATE AT
PREVIOUS MEASUREMENT OR EVENT TIME

TEVN I EVENT TIME

SUBROUTINES SUPPORTED: ERRANN

SUBROUTINES REQUIRED:	CORREL	DYNO	HYELS	JACOBI	GNAVM
	NTM	PSIM	STMPR	TITLE	TRAPAR
	GPRINT	MEAN	EIGHY		

LOCAL SYMBOLS BLANK DUMMY CALLING ARGUMENT

EGVCT ARRAY OF EIGENVECTORS CORRESPONDING TO THE
COLUMNS OF A GIVEN MATRIX

EGVL ARRAY OF EIGENVALUES RELATED TO THE
EIGENVECTORS CONTAINED IN EGVCT

EXTIJ INTERMEDIATE VARIABLE

ICODE INTERNAL CONTROL FLAG

OUT SQUARE ROOTS OF EIGENVALUES

PEIG MATRIX FOR WHICH HYPERELLIPSOID IS TO BE
COMPUTED

RF NOMINAL SPACECRAFT STATE AT EVENT TIME

VEIG MATRIX TO BE DIAGONALIZED

COMMON COMPUTED/USED: TRTM1 XF

COMMON COMPUTED: DELTM XI

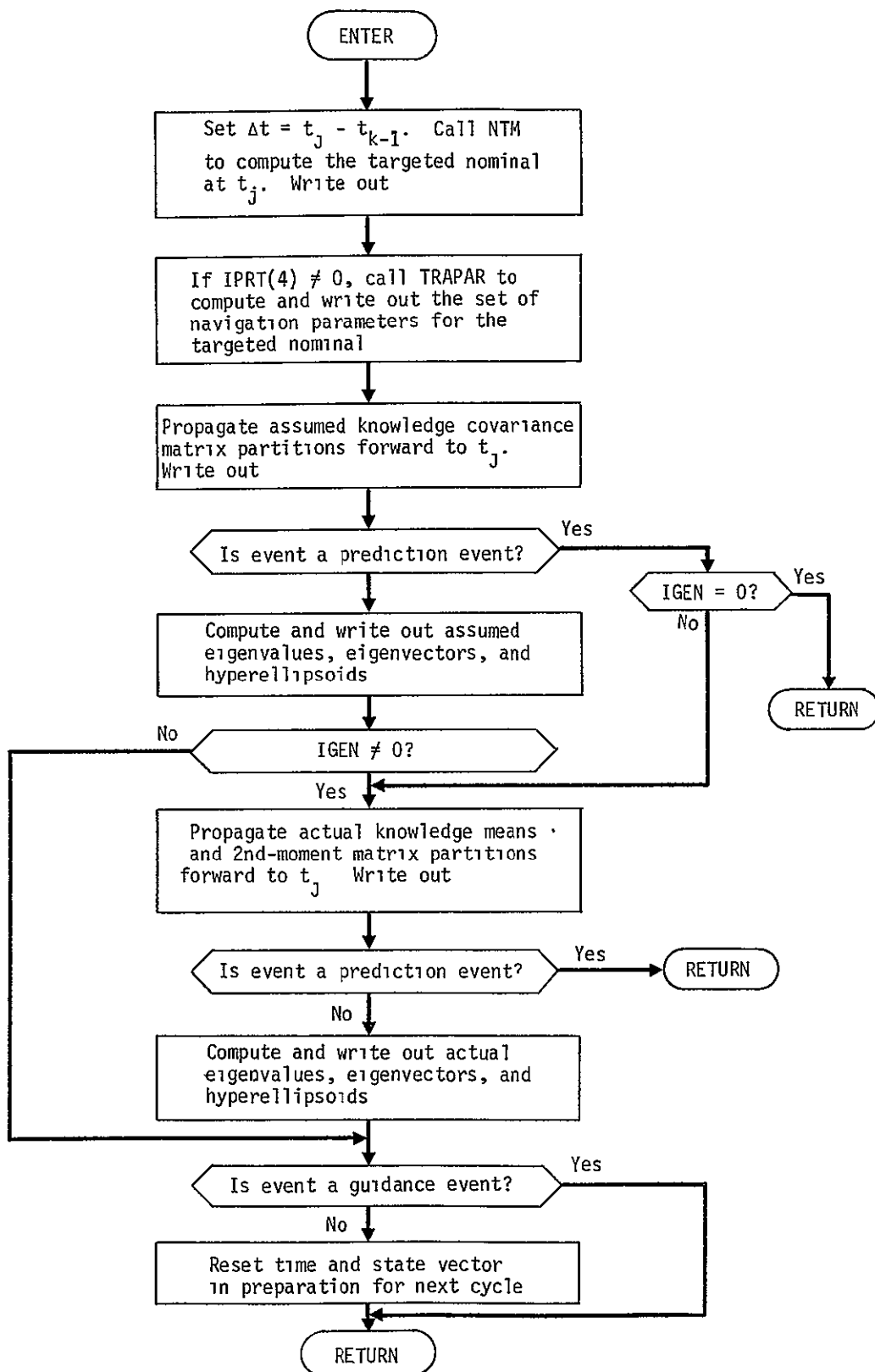
COMMON USED:	CXSU	CXSV	CXU	CXV	CXXS
	FOP	FOV	IEIG	IHYP1	IPRT
	ISTMC	NTMC	PS	P	Q
	UD	VG	XLAB		
	IGEN	GU	GV	G CXW	G CXSW
	GP	G CXXS	G CXU	G CXV	G PS
	G CXSU	G CXSV	QPR	RPR	EXT
	EXST				

SETEVN Analysis

Before executing any event in the error analysis/generalized covariance analysis program subroutine SETEVN is called to perform a series of computations that are common to all events. Subroutine SETEVN computes the targeted nominal trajectory at t_j and propagates the assumed and actual knowledge covariance partitions at t_{k-1} -- the time of the previous event or measurement -- forward to time t_j using the propagation equations found in subroutine GNAVM. The actual estimation error means are also propagated forward to t_j using the propagation equations found in subroutine MEAN.

For any event other than a prediction event, subroutine SETEVN also computes eigenvalues, eigenvectors, and hyperellipsoids of the position and velocity partitions of the assumed and actual knowledge covariance at t_j .

SETEVN Flow Chart



SUBROUTINE SETEVS

PURPOSE PERFORM ALL COMPUTATIONS COMMON TO MOST EVENTS IN THE
SIMULATION PROGRAM

CALLING SEQUENCE: CALL SETEVS(RI,TEVN,RI1,NCODE)

ARGUMENT: NCODE I EVENT CODE

RI I TARGETED NOMINAL SPACECRAFT STATE AT
PREVIOUS MEASUREMENT OR EVENT TIME

RI1 I MOST RECENT NOMINAL SPACECRAFT STATE AT
PREVIOUS MEASUREMENT OR EVENT TIME

TEVN I EVENT TIME

SUBROUTINES SUPPORTED: SIMULL

SUBROUTINES REQUIRED: CORREL DYNOS HYELS JACOBI NAVM
NTMS PSIM STMPR TITLES TRAPAR

LOCAL SYMBOLS DUM INTERMEDIATE VECTOR

EGVT ARRAY OF EIGENVECTORS

EGVL ARRAY OF EIGENVALUES

ICODE INTERNAL CONTROL FLAG

OUT SQUARE ROOTS OF EIGENVALUES

PEIG MATRIX WHOSE HYPERELLIPSOID IS TO BE
COMPUTED

RF1 MOST RECENT NOMINAL SPACECRAFT STATE AT
EVENT TIME

RF TARGETED NOMINAL SPACECRAFT STATE AT EVENT
TIME

VEIG MATRIX TO BE DIAGONALIZED

COMMON COMPUTED/USED: ADEVX EDEVX TRTH1 XF1 XF
XI1 ZF ZI

COMMON COMPUTED: DELTM XI

COMMON USED: ADEVXS CXSU CXSV CXU CXV
CXXS EDEVXS FOP FOV IEIG
IHYP1 IPRT ISTMC NDIH1 NGE

NQE	NTMC	PHI	PS	P
Q	TXXS	UO	VO	W
XLAB				

SETEVS Analysis

Prior to executing any event in the simulation mode, subroutine SETEVS is called to perform a series of computations which are common to all events. After computing the targeted nominal and most recent nominal states at the time of the event t_j , knowledge covariance partitions are propagated forward to time t_j from time t_{k-1} of the previous event or measurement using the prediction equations found in the NAVM Analysis section. The actual trajectory state at t_j is computed using

$$X_j = Z_j + \omega_j$$

where Z_j is the actual trajectory state assuming no unmodeled acceleration has been acting on the spacecraft, and ω_j is the contribution of the actual unmodeled acceleration to the actual trajectory state at t_j . The actual and predicted position/velocity deviations from the most recent nominal at t_j are given by

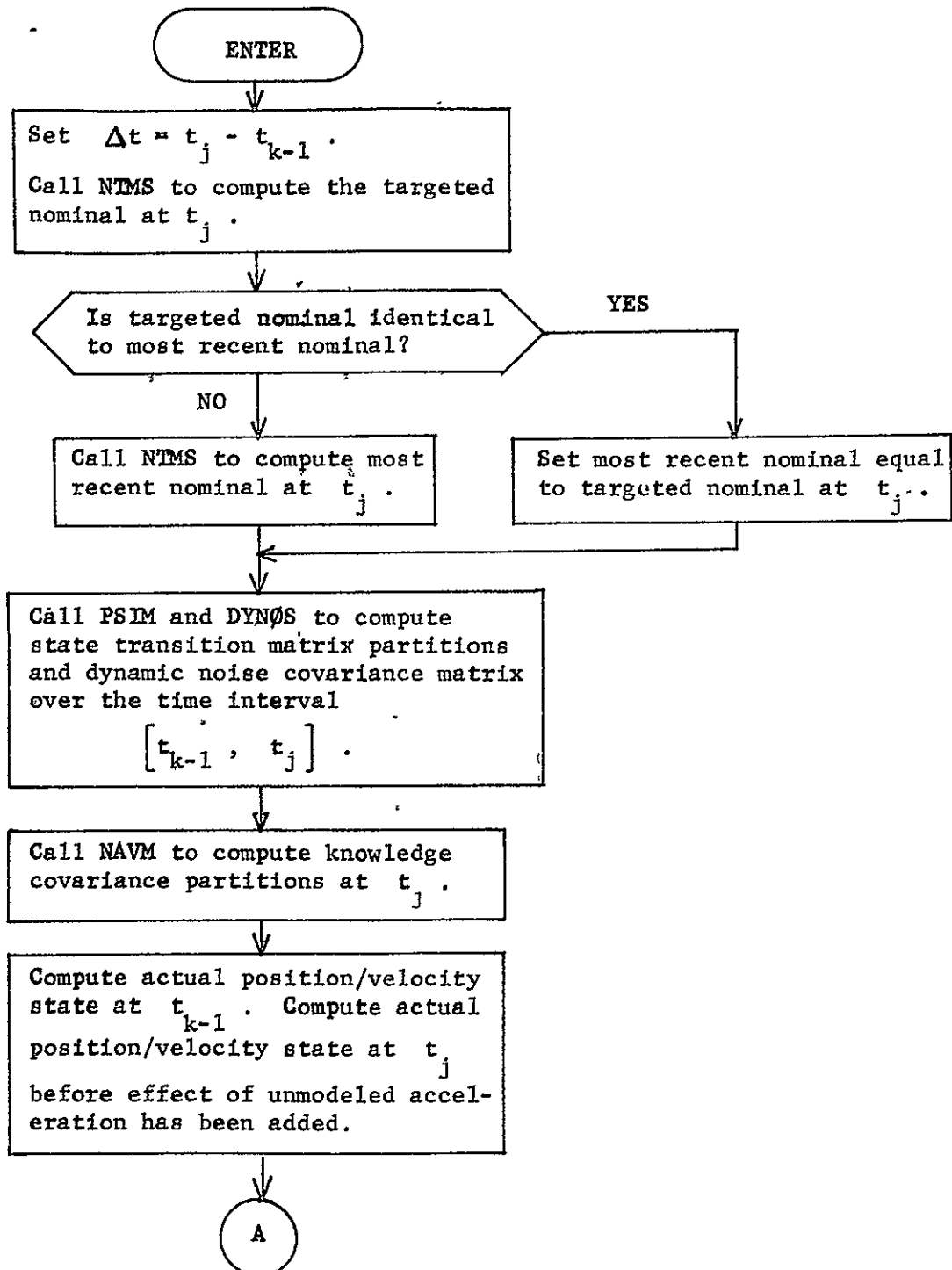
$$\delta \tilde{X}_j = X_j - \tilde{X}_j$$

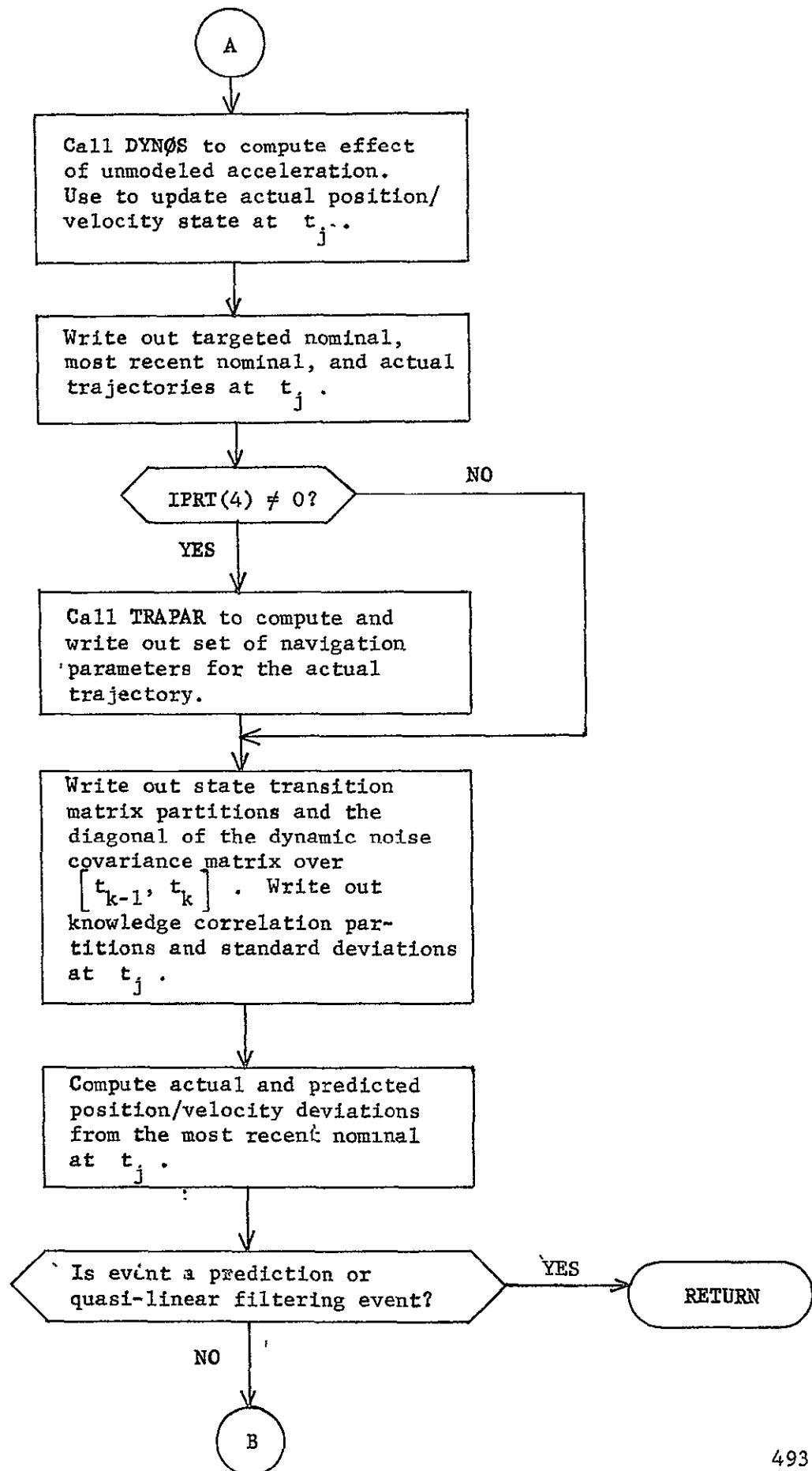
$$\text{and } \delta \tilde{X}_j = \Phi(t_j, t_{k-1}) \delta \tilde{X}_{k-1} + \theta_{xx_s}(t_j, t_{k-1}) \delta \tilde{X}_{s_j},$$

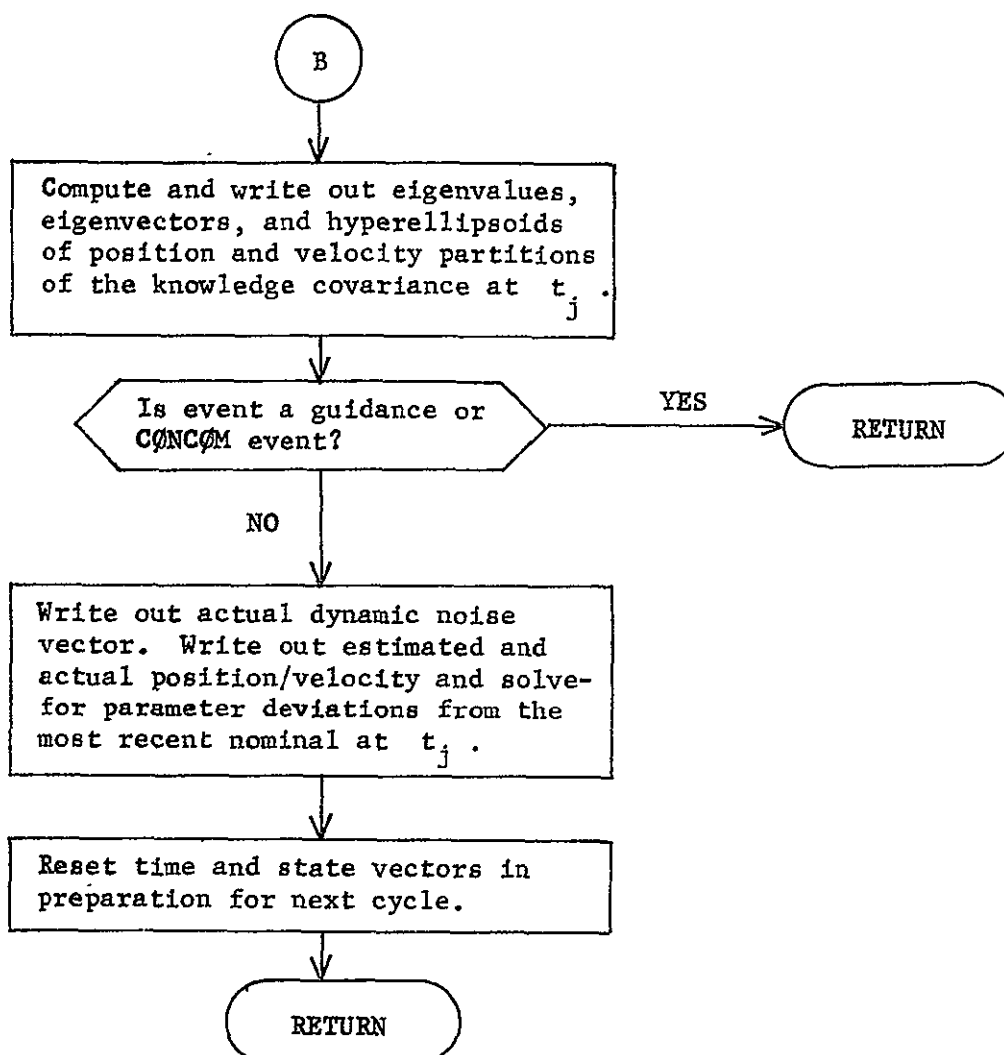
respectively, where Φ and θ_{xx_s} are the state transition matrix partitions over $[t_{k-1}, t_j]$.

For any event other than prediction and quasi-linear filtering events, subroutines SETEVS also computes eigenvalues, eigenvectors, and hyperellipsoids of the position and velocity partitions of the knowledge covariance at t_j .

SETEVS Flow Chart







PROGRAM SIMUL

PURPOSE: TO CONTROL THE COMPUTATIONAL FLOW THROUGH THE BASIC
CYCLE (MEASUREMENT PROCESSING) AND ALL EVENTS IN THE
SIMULATION PROGRAM

SUBROUTINES SUPPORTED: MAIN

SUBROUTINES REQUIRED:	BIAS	DYNOS	MENOS	NAVM	NTMS
	TRAKS	PRINT4	PSIM	SCHED	SETEVS

LOCAL SYMBOLS	BVAL	ACTUAL MEASUREMENT BIAS VECTOR
	DUMM	INTERMEDIATE VARIABLE
	DUM	INTERMEDIATE VECTOR
	IPRN	MEASUREMENT PRINT TIME COUNTER
	MMCODE	MEASUREMENT CODE
	NEVENT	EVENT COUNTER
	RNUM	RANDOM MEASUREMENT NOISE
	TRTH2	TIME OF THE MEASUREMENT

COMMON COMPUTED/USED:	ADEVX	ANGIS	AY	EDEVXS	EDEVX
	EY	ICODE	MCNTR	NAFC	RES
	RF1	RI1	RI	TEVN	TRTM1
	XF1	XF	XI1	XI	ZF
	ZI				

COMMON COMPUTED:	AYHEY	DELTH	EDEVSM	EDEVXM
------------------	-------	-------	--------	--------

COMMON USED:	AK	AM	AR	FNTM	H
	IEVNT	IPRINT	ISTMC	NAE	NAF6
	NDIM1	NEV	NGE	NMN	NQE
	NR	NTMC	PHI	RF	S
	TEV	TXXS	W	ZERO	

SIMUL Analysis

The primary function of subroutine SIMUL is to control the computational flow through the basic cycle (measurement processing) and all events in the simulation mode. Subroutine SIMUL also performs some computations in the basic cycle. All event-related analysis is presented in the event subroutines themselves and will not be treated below.

In the basic cycle the first task of SIMUL is to control the generation of targeted nominal and most recent nominal spacecraft states, \bar{X}_{k+1} and \tilde{X}_{k+1} , respectively, at time t_k , given states \bar{X}_k and \tilde{X}_k at time t_k . Then, calling PSIM, DYNØS, TRAKS, and MENØS, successively, SIMUL controls the computation of all matrix information required by subroutine NAVM in order to compute the covariance matrix partitions at time t_{k+1}^+ immediately following the measurement.

After computing the actual state X_k at time t_k from

$$X_k = \tilde{X}_k + \delta \tilde{X}_k \quad (1)$$

where $\delta \tilde{X}_k$ is the actual spacecraft state deviation from the most recent nominal, SIMUL controls the generation of the actual state Z_{k+1} at time t_k before the effect of unmodeled acceleration has been added. Then, having called DYNØS to compute the effect of unmodeled acceleration ω_{k+1} , SIMUL computes the actual state and actual state deviation at time t_{k+1} :

$$X_{k+1} = Z_{k+1} + \omega_{k+1} \quad (2)$$

$$\delta \tilde{X}_{k+1} = X_{k+1} - \tilde{X}_{k+1} \quad (3)$$

With both the most recent nominal and actual spacecraft states available at t_{k+1} , SIMUL calls TRAKS twice in succession to compute the ideal measurements \tilde{Y}_{k+1} and Y_{k+1} , respectively, which would be made at each of these trajectory states. Calling MENØS, RNUM, and BIAS to compute the actual measurement noise and bias corrupting the ideal measurement associated with the actual state, SIMUL computes the actual measurement at time t_{k+1} using

$$y_{k+1}^a = \tilde{y}_{k+1} + b_{k+1} + \nu_{k+1} \quad (4)$$

where b_{k+1} and ν_{k+1} represent the actual measurement bias and noise, respectively.

All information required for computing both predicted and filtered state deviations from the most recent nominal at t_{k+1} is now available. With Φ and θ_{xx_s} denoting state transition matrix partitions over the time interval $[t_k, t_{k+1}]$, SIMUL computes the predicted spacecraft state deviations and solve-for parameter deviations at t_{k+1} using

$$\tilde{\delta X}_{k+1}^- = \Phi \tilde{\delta X}_k^+ + \theta_{xx_s} \tilde{\delta X}_{s_k}^+ \quad (5)$$

$$\tilde{\delta X}_{s_{k+1}}^- = \tilde{\delta X}_{s_k}^+ \quad (6)$$

Prior to computing filtered deviations, SIMUL computes the measurement residual from

$$\epsilon_{k+1} = (Y_{k+1}^a - \tilde{Y}_{k+1}) - H_{k+1} \tilde{\delta X}_{k+1}^- - M_{k+1} \tilde{\delta X}_{s_{k+1}}^- \quad (7)$$

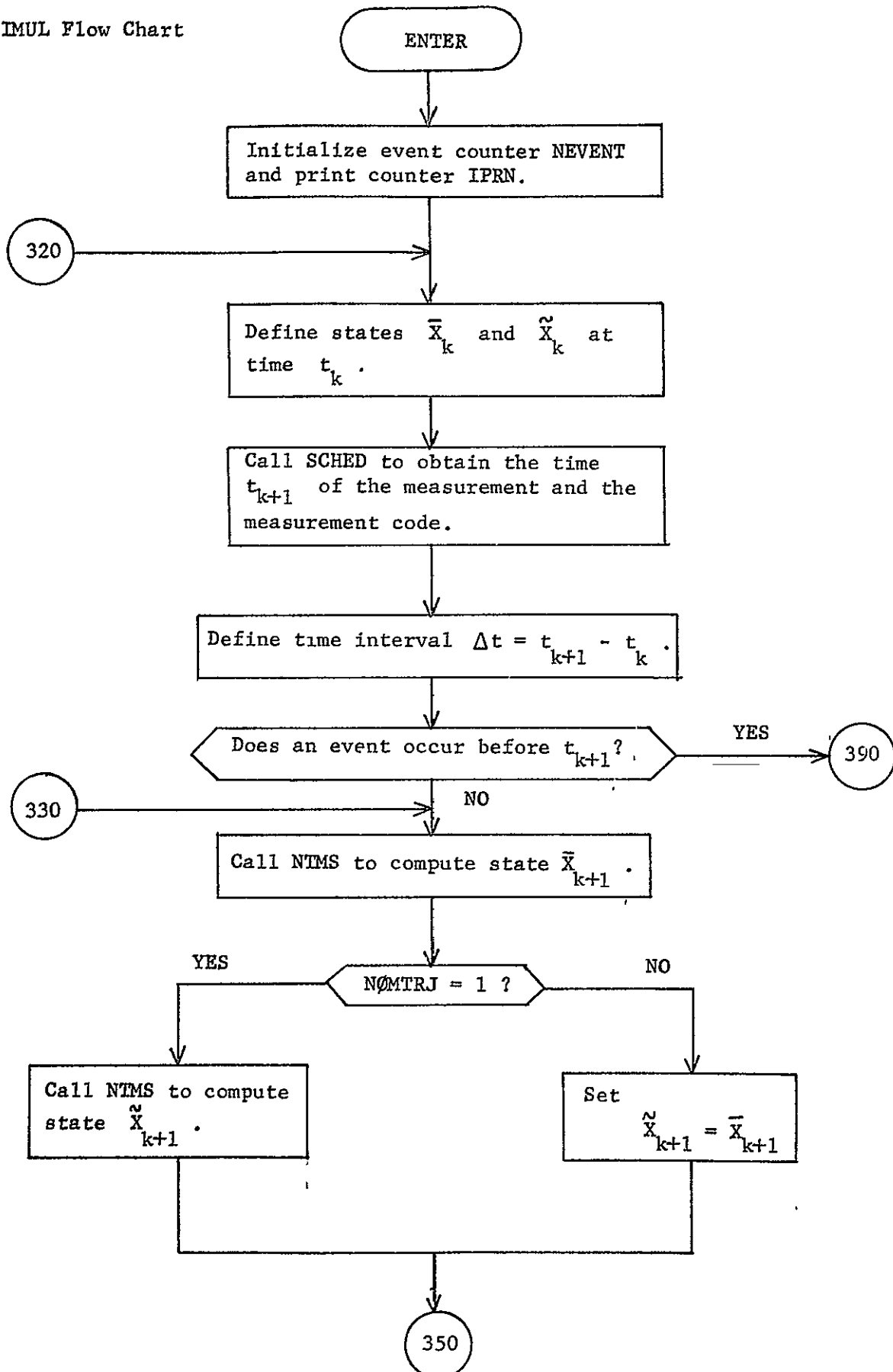
where H_{k+1} and M_{k+1} are observation matrix partitions. Filtered spacecraft state deviations and solve-for parameter deviations are then computed from

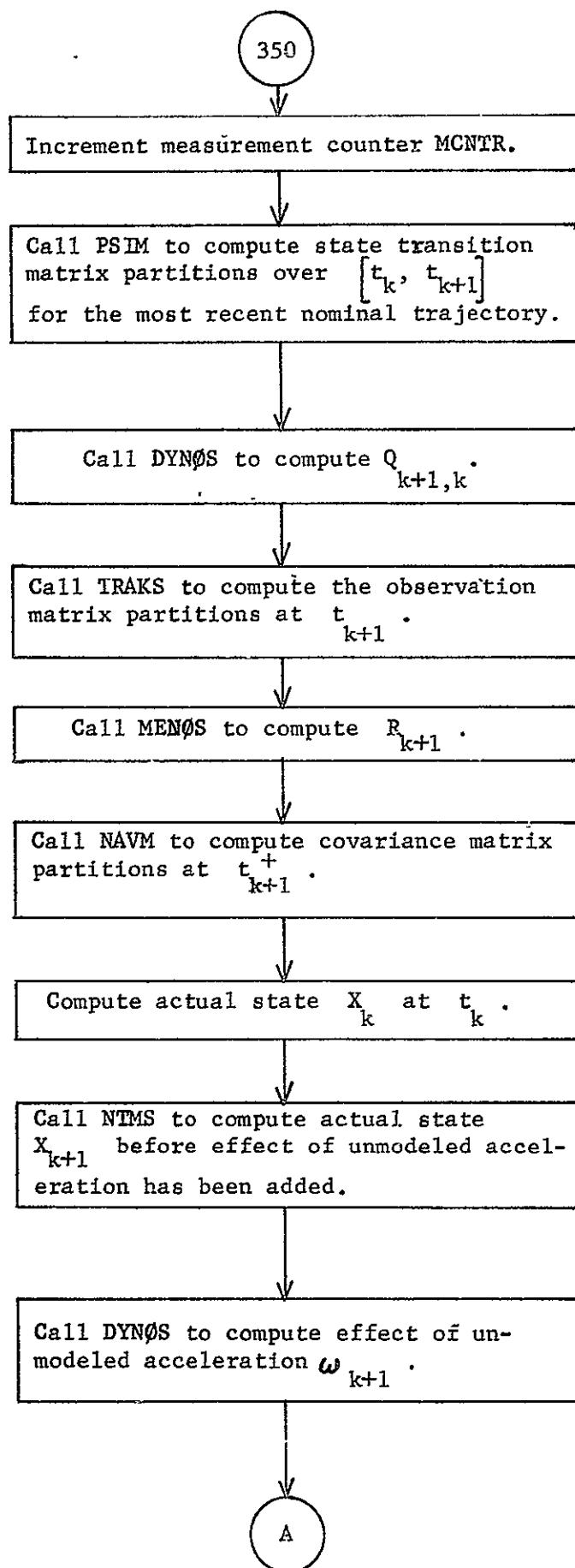
$$\tilde{\delta X}_{k+1}^+ = \tilde{\delta X}_{k+1}^- + K_{k+1} \epsilon_{k+1} \quad (8)$$

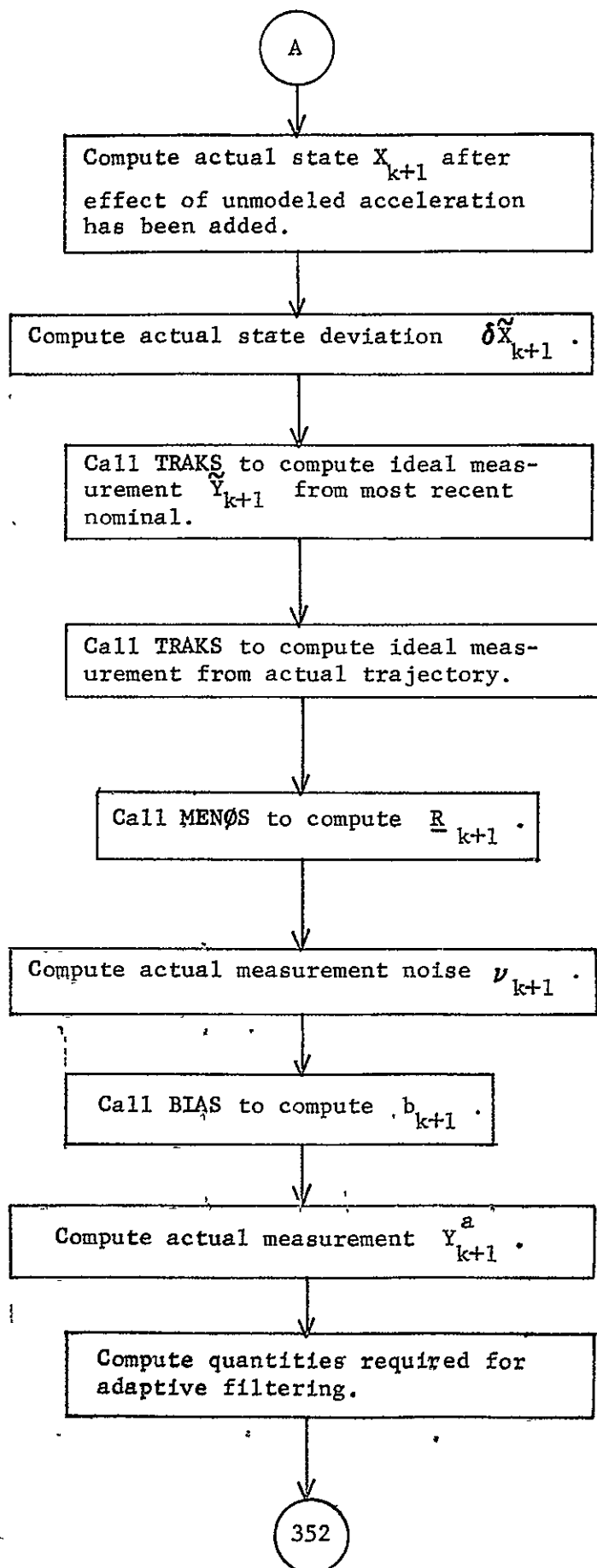
$$\tilde{\delta X}_{s_{k+1}}^+ = \tilde{\delta X}_{s_{k+1}}^- + S_{k+1} \epsilon_{k+1} \quad (9)$$

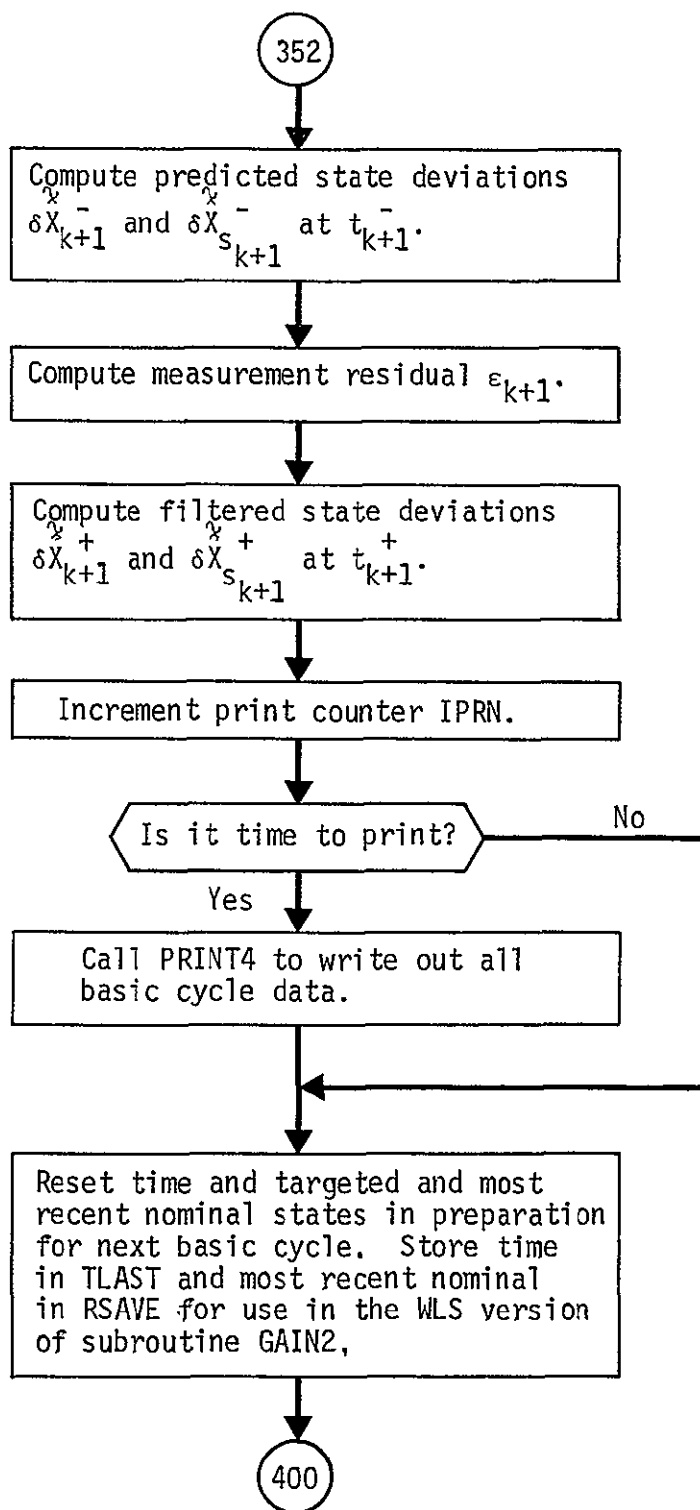
where K_{k+1} and S_{k+1} are the filter gain constants.

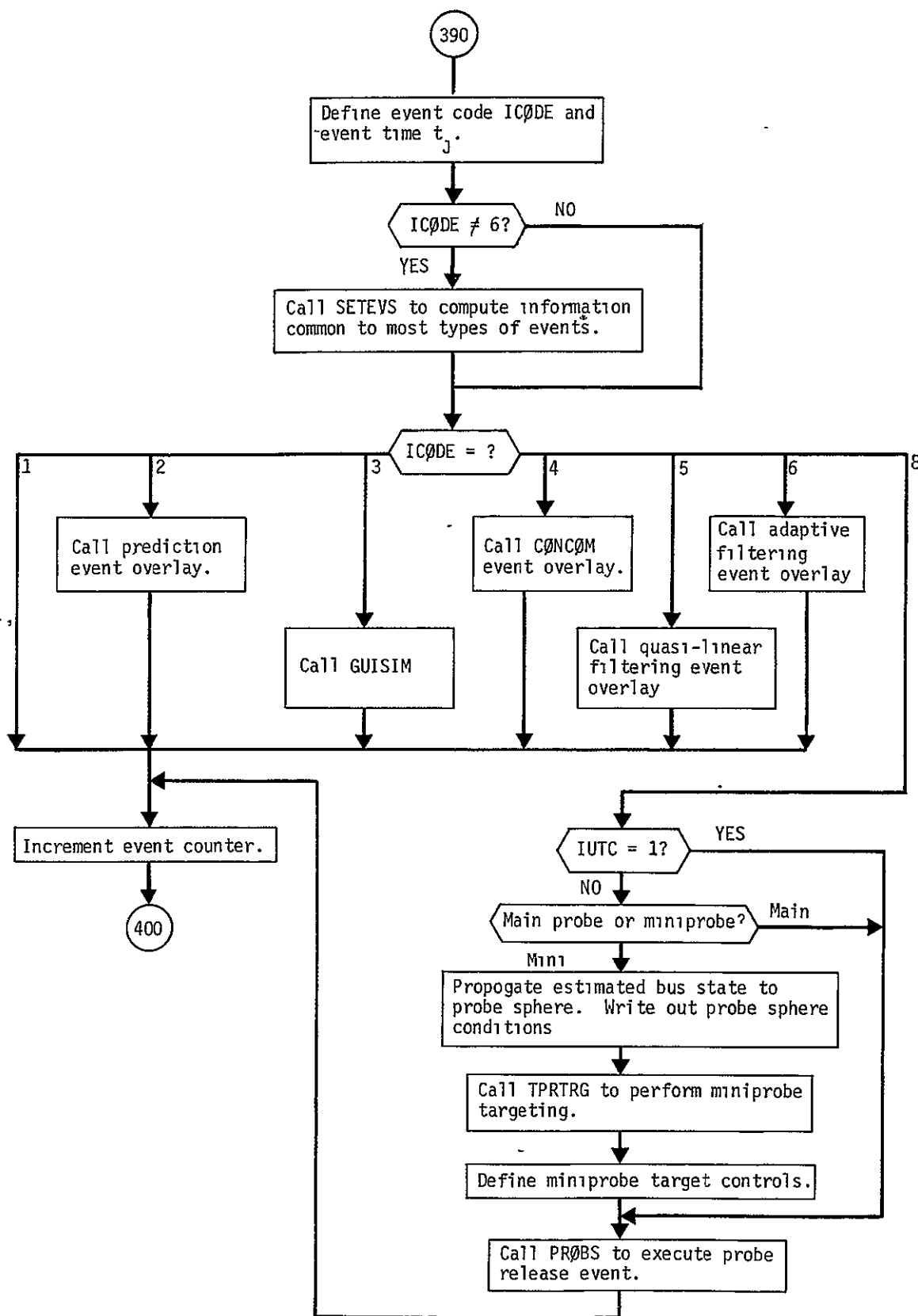
SIMUL Flow Chart

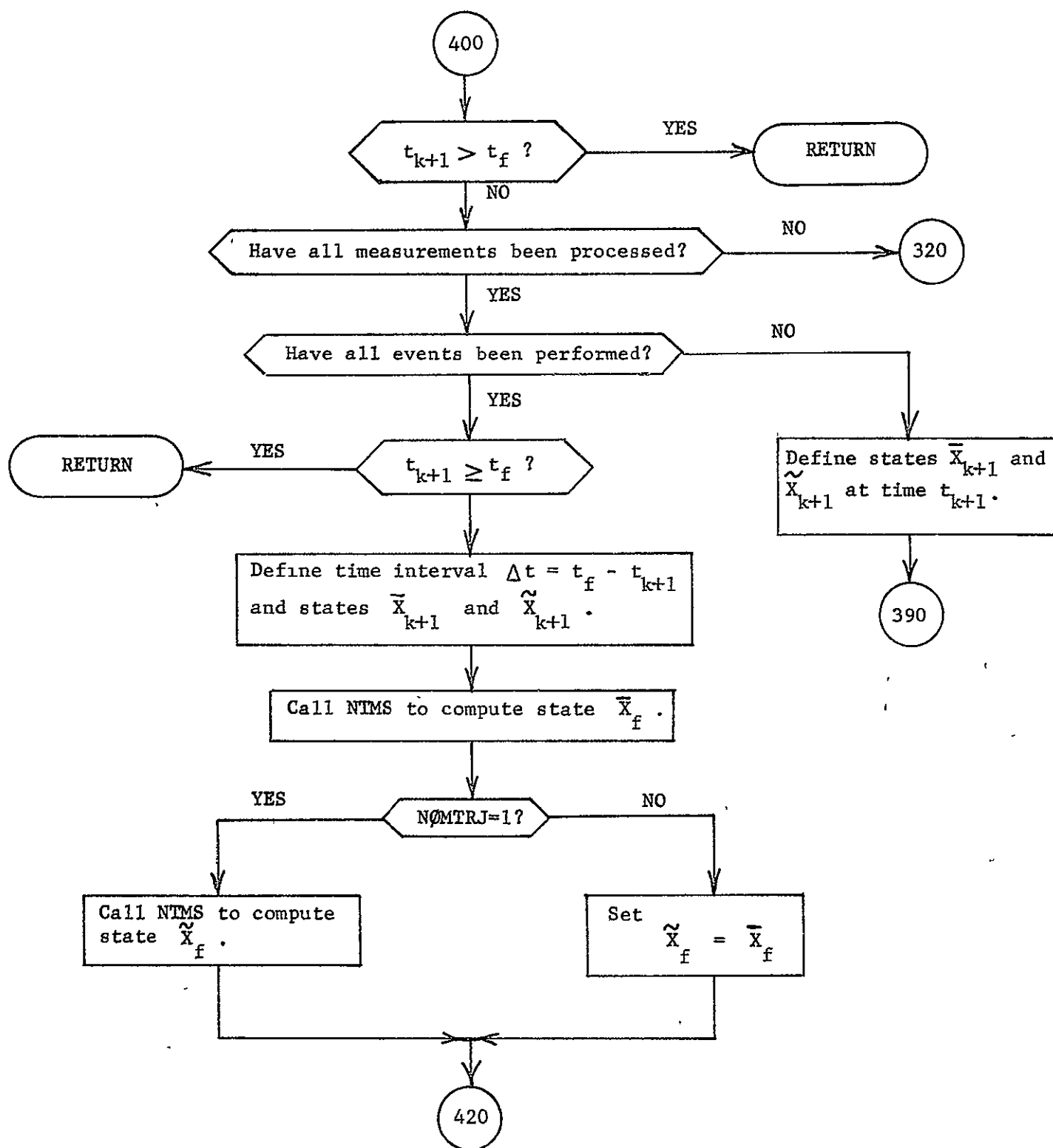


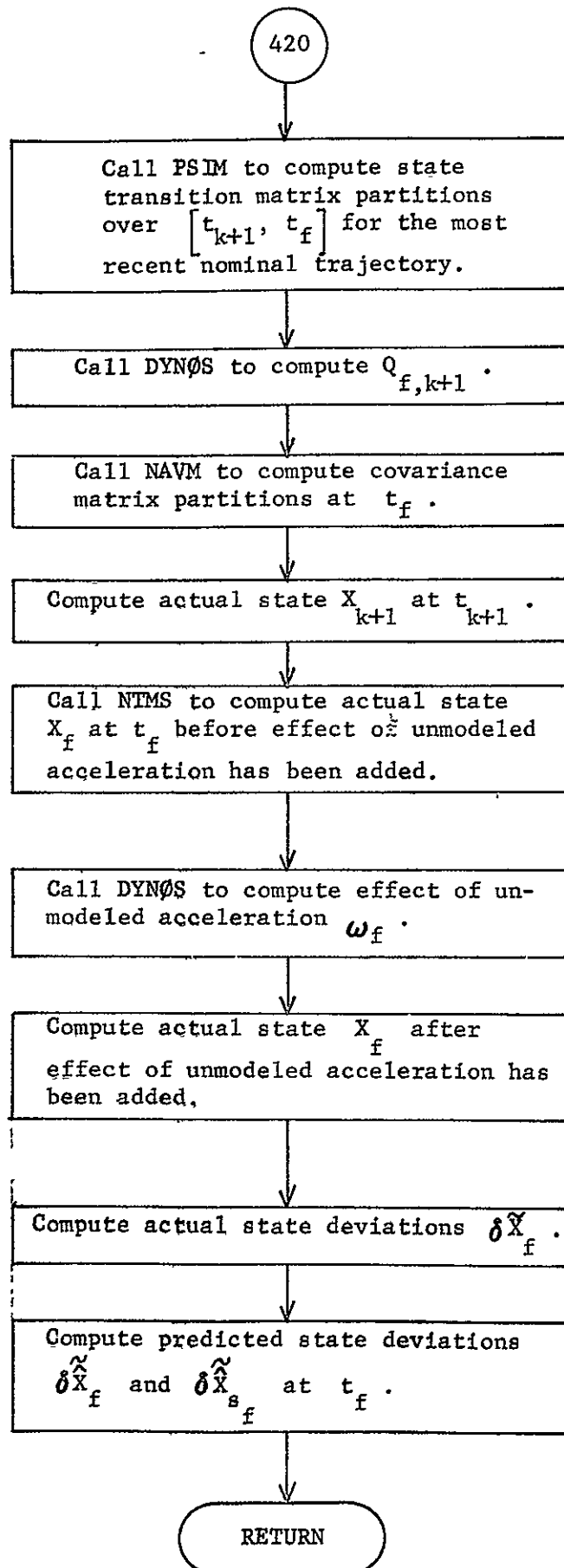












SUBROUTINE SKEDM

PURPOSE: TO SET UP THE MEASUREMENT SCHEDULE USED INTERNALLY

CALLING SEQUENCE: CALL SKEDM(NENT, TMN, MCODE, NMN, FNTM, INUM, IFLG)

ARGUMENTS: NENT I NUMBER OF MEASUREMENT CARDS
TMN O VECTOR OF TIMES OF MEASUREMENTS
MCODE O VECTOR OF MEASUREMENT CODES CORRESPONDING
TO ENTRIES IN TMN
NMN O MAXIMUM NUMBER OF ENTRIES IN TMN VECTOR
FNTM I FINAL TRAJECTORY TIME
INUM I DIMENSION OF TMN AND MCODE VECTORS
IFLG I INDEX FOR PRINTING TITLE

SUBROUTINES SUPPORTED: DATA1 DATA1S

LOCAL SYMBOLS: AMIN INTERMEDIATE TIME VARIABLE
AP INTERMEDIATE TIME ARRAY
EP50 10. TO 50TH POWER
ICNT COUNTER
IROW ROW INDEX
MEAS MEASUREMENT CODE ARRAY
SCHED MEASUREMENT TIME ARRAY
XLAB LABEL

SUBROUTINE SOIPS

PURPOSE: TO CONICALLY EXTRAPOLATE FROM NEAREST INTEGRATION STATE TO
OBTAIN IMPACT DATA AT SOI AND AT PLANET SURFACE

ARGUMENT: ISOIPS I FLAG INDICATION OPERATING MODE OF SUBROUTINE
= 1 EXTRAPOLATE FOR SOI IMPACT DATA
= 2 EXTRAPOLATE FOR PLANET IMPACT DATA

RSPHM 0 MAGNITUDE OF PLANETOCENTRIC POSITION VECTOR OF
TRAJECTORY AT SPHERE OF INTEREST IN KM
PLANETOCENTRIC ECLIPTIC

RSPH 0 PLANETOCENTRIC ECLIPTIC POSITION VECTOR OF
TRAJECTORY AT SPHERE OF INTEREST IN KM

TMU I GRAVITATIONAL CONSTANT OF TARGET PLANET IN
KM**3/SEC**2

VSPHM 0 MAGNITUDE OF PLANETOCENTRIC VELOCITY OF TRAJECTORY
AT SPHERE OF INTEREST IN KM/SEC

VSPH 0 PLANETOCENTRIC ECLIPTIC VELOCITY VECTOR OF
TRAJECTORY AT SPHERE OF INTEREST IN KM/SEC

SUBROUTINES SUPPORTED: VMP

SUBROUTINES REQUIRED: CAREL ELCAR PECEQ SUBSOL

LOCAL SYMBOLS: AS SEMIMAJOR AXIS OF PLANETOCENTRIC OSCULATING CONIC
IN KM

CSTAS COSINE OF TRUE ANOMALY OF PLANETOCENTRIC
OSCULATING CONIC AT SPHERE OF INTEREST

DJEPOC JULIAN DATE EPOCH 1900 WHEN TRAJECTORY PIERCES
SPHERE OF INTEREST

DTNIS TIME INTERVAL IN SECONDS ALONG PLANETOCENTRIC
OSCULATING CONIC FROM NEAREST INTEGRATION STATE
INSIDE SPHERE OF INTEREST TO SPHERE ITSELF

DTPNI TIME INTERVAL IN SECONDS ALONG PLANETOCENTRIC
OSCULATING CONIC FROM PERAPSIS TO NEAREST
INTEGRATION STEP INSIDE SPHERE OF INTEREST

DTPS TIME INTERVAL IN SECONDS ALONG PLANETOCENTRIC
OSCULATING CONIC FROM PERIAPSIS TO SPHERE OF
INTEREST

DTS CONVERSION FACTOR FROM DAYS TO SECONDS

ES ECCENTRICITY OF PLANETOCENTRIC OSCULATING CONIC

PAS PLANETOCENTRIC ECLIPTIC UNIT VECTOR POINTING IN
DIRECTION OF PERIAPSIS OF PLANETOCENTRIC
OSCULATING CONIC

PS SEMILATUS RECTUM OF PLANETOCENTRIC OSCULATING
CONIC IN KM

QAS PLANETOCENTRIC ECLIPTIC UNIT VECTOR LYING IN
ORBITAL PLANE OF PLANETOCENTRIC OSCULATING CONIC 9
DEGREES ADVANCED FROM PA

RS RADIUS OF SPHERE OF INTEREST IN KM

RTO CONVERSION FACTOR FROM RADIANS TO DEGREES

SNTAS SINE OF TRUE ANAMOLY OF PLANETOCENTRIC OSCULATING
CONIC AT SPHERE OF INTEREST

TAS TRUE ANAMOLY OF PLANETOCENTRIC OSCULATING CONIC
AT NEAREST INTEGRATION STATE INSIDE SPHERE OF
INTEREST

TRANSF TRANSFORMATION MATRIX FROM ECLIPTIC TO PROBE-
SPHERE COORDINATES

WAS PLANETOCENTRIC ECLIPTIC UNIT VECTOR POINTING IN
DIRECTION OF ANGULAR MOMENTUM OF PLANETOCENTRIC
OSCULATION CONIC

WS ECLIPTIC ARGUMENT OF PERIAPSIS IN DEGREES OF
PLANETOCENTRIC OSCULATING CONIC

XIS ECLIPTIC INCLINATION IN DEGREES OF PLANETOCENTRIC
OSCULATING CONIC

XNS ECLIPTIC RIGHT ASCENSION OF ASCENDING NODE IN
DEGREES OF PLANETOCENTRIC OSCULATING CONIC

COMMON COMPUTED/USED: DEPOC D

COMMON COMPUTED: DCIMP RAIMP

COMMON USED: ALNGTH IPCSP IP NTP ONE
RADIUS RPSP SPHERE

SUBROUTINE SPHIMP

PURPOSE: TO CALCULATE TRUE ANAMOLY AND TIME FROM PERIAPSIS AT WHICH CONI
APPROACH TRAJECTORY PIERCES PLANETOCENTRIC SPHERE OF GIVEN
RADIUS

ARGUMENT: A I SEMIMAJOR AXIS OF CONIC IN KM
CSTAI 0 COSINE OF TRUE ANAMOLY AT SPHERE OF INTEREST
DTPI 0 TIME INTERVAL IN SECONDS ON CONIC FROM PERIAPSIS T
SPHERE OF INTEREST
E I ECCENTRICITY OF CONIC
GMUP I GRAVITATIONAL CONSTANT OF PLANET IN KM^3/SEC^2
RS I RADIUS OF SPHERE OF INTEREST IN KM
SNTAI 0 SINE OF TRUE ANAMOLY AT SPHERE OF INTEREST

SUBROUTINES SUPPORTED: TPPROP TPRTRG |

SUBROUTINES REQUIRED: HYPT

LOCAL SYMBOLS: ONE CONSTANT 1.

ORBH RECIPROCAL OF MEAN ORBITAL RATE IN SEC/RADIANS

P SEMILATUS RECTUM OF CONIC IN KM

SUBROUTINE SPACE

PURPOSE: COUNTS THE NUMBER OF LINES BEING PRINTED TO DETERMINE
WHEN TO SKIP TO THE NEXT PAGE WITH A NEW HEADING

CALLING SEQUENCE CALL SPACE(LINES)

ARGUMENT LINES I NUMBER OF LINES THAT WILL BE WRITTEN IN
 THE NEXT OUTPUT STATEMENT

SUBROUTINES SUPPORTED: INPUTZ PRINT VECTOR VMP

SUBROUTINES REQUIRED: NEWPGE

COMMON COMPUTED/USED: LINCT

COMMON USED: LINPGE

SUBROUTINE STAPRL

PURPOSE: TO COMPUTE THE PARTIAL DERIVATIVES OF STATION LOCATION ERRORS.

CALLING SEQUENCE: CALL STAPRL(AL,ALON,ALAT,PAT2,VEC,PA)

ARGUMENT: AL I ALTITUDE OF THE STATION
 ALAT I LATITUDE OF THE STATION
 ALON I LONGITUDE OF THE STATION
 PA O PARTIAL OF STATION POSITION AND VELOCITY
 WITH RESPECT TO ALTITUDE, LATITUDE AND
 LONGITUDE
 PAT2 I LONGITUDE + OMEGA*(CURRENT TIME-LAUNCH
 TIME)
 VEC UNUSED

SUBROUTINES SUPPORTED: TRAKS TRAKM

LOCAL SYMBOLS: G1 SINE OF LATITUDE
 G2 COSINE OF LATITUDE
 G3 SINE(PHI + OMEGA(T-UNIVT))
 G4 COSINE(PHI + OMEGA(T-UNIVT))
 WHERE PHI = LONGITUDE
 OMEGA = EARTH ROTATION RATE
 T = TIME
 UNIVT = UNIVERSAL TIME
 G5 SINE OF OBLIQUITY OF EARTH
 G6 COSINE OF OBLIQUITY OF EARTH
 OMEG OMEGA IN PROPER UNITS

COMMON USED: EPS OMEGA TM

STAPRL Analysis

The ecliptic components of the position and velocity of a tracking station relative to the Earth are related to station location parameters R , θ , and ϕ through the following set of equations:

$$X_s = R \cos \theta \cos G$$

$$Y_s = R \cos \theta \cos \epsilon \sin G + R \sin \theta \sin \epsilon$$

$$Z_s = -R \cos \theta \sin \epsilon \sin G + R \sin \theta \cos \epsilon$$

$$\dot{X}_s = -\omega R \cos \theta \sin G$$

$$\dot{Y}_s = \omega R \cos \theta \cos \epsilon \cos G$$

$$\dot{Z}_s = -\omega R \cos \theta \sin \epsilon \cos G$$

where $G = \phi + \omega(t - T)$, and T is the universal time at some epoch (usually launch time).

Subroutine STAPRL computes the negative of the partials of the previous quantities with respect to the station location parameters R , θ , and ϕ . These partials are summarized below:

$$-\frac{\partial X_s}{\partial R} = -\cos \theta \cos G$$

$$-\frac{\partial X_s}{\partial \theta} = R \sin \theta \cos G$$

$$-\frac{\partial X_s}{\partial \phi} = R \cos \theta \sin G$$

$$-\frac{\partial Y_s}{\partial R} = -[\sin \epsilon \sin \theta + \cos \epsilon \cos \theta \sin G]$$

$$-\frac{\partial Y_s}{\partial \theta} = R \cos \epsilon \sin \theta \sin G - R \sin \epsilon \cos \theta$$

$$-\frac{\partial Y_s}{\partial \phi} = -R \cos \epsilon \cos \theta \cos G$$

$$-\frac{\partial Z_s}{\partial R} = \sin \epsilon \cos \theta \sin G - \cos \epsilon \sin \theta$$

$$- \frac{\partial Z_s}{\partial \theta} = - [R \sin \epsilon \sin \theta \sin G + R \cos \epsilon \cos \theta]$$

$$- \frac{\partial Z_s}{\partial \phi} = R \sin \epsilon \cos \theta \cos G$$

$$- \frac{\partial \dot{X}_s}{\partial R} = \omega \cos \theta \sin G$$

$$- \frac{\partial \dot{X}_s}{\partial \theta} = - \omega R \sin \theta \sin G$$

$$- \frac{\partial \dot{X}_s}{\partial \phi} = \omega R \cos \theta \cos G$$

$$- \frac{\partial \dot{Y}_s}{\partial R} = - \omega \cos \theta \cos \epsilon \cos G$$

$$- \frac{\partial \dot{Y}_s}{\partial \theta} = \omega R \cos \epsilon \sin \theta \cos G$$

$$- \frac{\partial \dot{Y}_s}{\partial \phi} = \omega R \cos \epsilon \cos \theta \sin G$$

$$- \frac{\partial \dot{Z}_s}{\partial R} = \omega \sin \epsilon \cos \theta \cos G$$

$$- \frac{\partial \dot{Z}_s}{\partial \theta} = - \omega R \sin \epsilon \sin \theta \cos G$$

$$- \frac{\partial \dot{Z}_s}{\partial \phi} = - \omega R \sin \epsilon \cos \theta \sin G$$

SUBROUTINE STIMP

PURPOSE: CALCULATE B-PLANE ASYMPTOTE PIERCE-POINT COORDINATES
IN IMPACT PLANE OF A HYPERBOLA, GIVEN THE STATE

ARGUMENT: A I SEMI-MAJOR AXIS OF HYPERBOLA IN KM
 BDR O PROJECTION OF BV ON RV (B.R) IN KM
 BOT O PROJECTION OF BV ON TV (B.T) IN KM
 B O MAGNITUDE OF BV IN -KM
 GMX I GRAVITATIONAL CONSTANT OF PLANET
 RV O PLANETOCENTRIC ECLIPTIC UNIT VECTOR
 IN DIRECTION OF CROSS PRODUCT OF
 HYPERBOLA ASYMPTOTE BY ECLIPTIC POLE VECTOR
 R I PLANETOCENTRIC ECLIPTIC POSITION VECTOR
 OF GIVEN HYPERBOLA STATE IN KM
 SV O PLANETOCENTRIC ECLIPTIC UNIT VECTOR
 IN DIRECTION OF HYPERBOLA ASYMPTOTE
 TV O CROSS PRODUCT OF SV AND RV
 V PLANETOCENTRIC ECLIPTIC VELOCITY VECTOR
 OF GIVEN HYPERBOLA STATE IN KM/SEC

SUBROUTINES SUPPORTED: TPPROP TPRTRG

SUBROUTINES REQUIRED: USCALE UXV

LOCAL SYMBOLS: AB PRODUCT OF HYPERBOLA ECCENTRICITY AND
 MAGNITUDE OF SEMI-MAJOR AXIS IN KM
 BV PLANETOCENTRIC ECLIPTIC VECTOR IN KM
 TO HYPERBOLA ASYMPTOTE PIERCE POINT
 IN IMPACT PLANE
 CTA COSINE OF TRUE ANOMALY OF GIVEN STATE
 C1 MAGNITUDE OF HYPERBOLA ANGULAR MOMENTUM
 DUM MAGNITUDE OF CROSS PRODUCT OF SV AND
 ECLIPTIC POLE VECTOR
 E ECCENTRICITY OF HYPERBOLA
 ONE CONSTANT = 1.
 PV PLANETOCENTRIC ECLIPTIC UNIT VECTOR IN

DIRECTION OF HYPERBOLA PERIAPSIS

P	SEMI-LATUS RECTUM OF HYPERBOLA IN KM
QV	PLANETOCENTRIC ECLIPTIC UNIT VECTOR LYING IN HYPERBOLA PLANE AND ADVANCED 90 DEGREES FROM PA
RD	TIME RATE OF CHANGE OF HYPERBOLA RADIUS AT GIVEN STATE
RM	MAGNITUDE OF PLANETOCENTRIC POSITION
RRD	PRODUCT OF HYPERBOLA RADIUS AND ITS TIME RATE OF CHANGE OF GIVEN STATE
STA	SINE OF TRUE ANOMALY
TWO	CONSTANT = 2.
VS	SQUARED MAGNITUDE OF PLANETOCENTRIC VELOCITY OF GIVEN STATE
WV	PLANETOCENTRIC ECLIPTIC UNIT VECTOR IN DIRECTION OF HYPERBOLA ANGULAR MOMENTUM
ZERO	CONSTANT = 0.
Z	PLANETOCENTRIC ECLIPTIC UNIT VECTOR LYING IN HYPERBOLA PLANE AND 90 DEGREES ADVANCED FROM POSITION VECTOR

STIMP Analysis

Subroutine STIMP converts a planetocentric ecliptic state vector $(\underline{r}/\underline{v})^T$ to the more readily targetable impact plane coordinates $B \cdot T$ and $B \cdot R$. These coordinates are preferred as target variables for two basic reasons: (1) they generally exhibit reasonably linear dependence on the targeting controls, and (2) in probe targeting they obviate the need for defining a pseudoimpact point when the probe misses the planet in an early iteration.

The impact coordinates are defined in terms of the direction of the trajectory hyperbolic excess velocity, \underline{v}_∞ , and the north ecliptic pole vector, \underline{K} . Let

$$\underline{S} = \underline{v}_\infty \quad \underline{v}_\infty \quad (1)$$

$$\underline{T} = \frac{\underline{S} \times \underline{K}}{\|\underline{S} \times \underline{K}\|} \quad (2)$$

$$\underline{R} = \underline{S} \times \underline{T} \quad (3)$$

where all the vectors are assumed to originate at the planet center. Thus \underline{T} , \underline{R} and \underline{S} form a right-handed Cartesian frame with \underline{S} pointing in the direction of the hyperbolic excess velocity and \underline{T} lying in the ecliptic plane. The plane containing the vector \underline{T} and \underline{R} is known as the impact plane. \underline{B} is defined as that unique vector from the planet center to the point where the trajectory asymptote pierces the impact plane. $B \cdot T$ and $B \cdot R$ are then simply the components of \underline{B} along the T and R axes, respectively. Figure 1 illustrates all of these terms for the case of a single probe trajectory.

In addition to calculating $B \cdot T$ and $B \cdot R$, STIMP also makes available other approach trajectory parameters useful in the STEAP auxiliary targeting scheme. These are \underline{S} , \underline{T} , \underline{R} , and \underline{B} in the inertial ecliptic frame as well as the approach hyperbola semi-major axis, a .

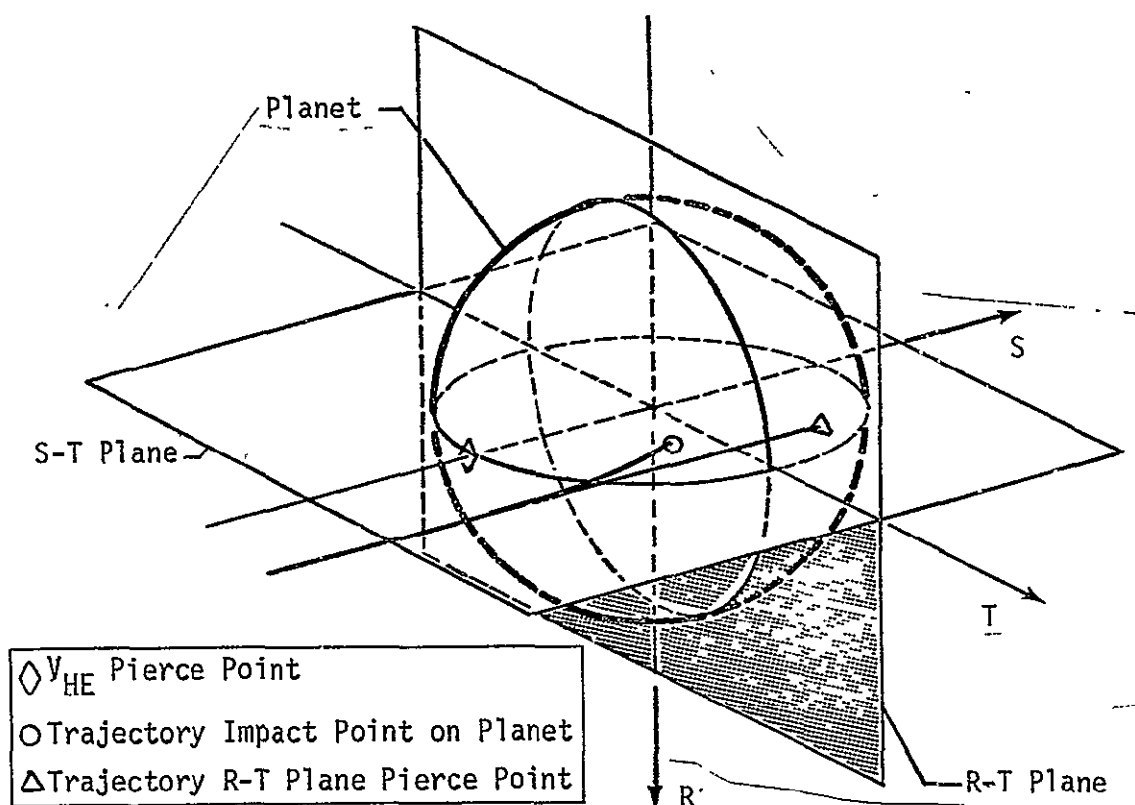


Figure 1 Single-Probe Auxiliary Targeting Illustration

The formulae used in calculating these elements are well known results derived in most engineering treatments of two-body motion (see, for example, Ref 2). Hence they will simply be listed here in the order they are used. The standard conic symbolism is used throughout:

$$\underline{h} = \underline{r} \times \underline{v} \quad (4)$$

$$\underline{W} = \underline{h}/h \quad (5)$$

$$\dot{\underline{r}} = \underline{r} \cdot \underline{v}/r \quad (6)$$

$$p = h^2/\mu \quad (7)$$

$$a = r/(2 - rv^2/\mu) \quad (8)$$

$$e = \sqrt{1 - p/a} \quad (9)$$

$$\cos \theta = (p - r)/er \quad (10)$$

$$\sin \theta = \dot{r}h/e\mu \quad (11)$$

$$B = \sqrt{p|a|} \quad (12)$$

$$\underline{Z} = (\underline{r}\underline{v} - \dot{\underline{r}}\underline{r})/h \quad (13)$$

$$\underline{P} = \frac{\underline{r}}{r} \cos \theta - \underline{Z} \sin \theta \quad (14)$$

$$\underline{Q} = \frac{\underline{r}}{r} \sin \theta + \underline{Z} \cos \theta \quad (15)$$

$$\underline{S} = \underline{P}/e + \frac{Q}{e} \sqrt{e^2 - 1} \quad (16)$$

$$\underline{B} = p\underline{P}/e + \frac{aQ}{e} \sqrt{e^2 - 1} \quad (17)$$

$$\underline{T} = \left(s_2^2, -s_1^2, 0 \right)^T / \sqrt{s_1^2 + s_2^2} \quad (18)$$

$$\underline{R} = \left(-s_3T_2, s_3T_1, s_1T_2 - s_2T_1 \right)^T \quad (19)$$

$$B \cdot T = B_1T_1 + B_2T_2 \quad (20)$$

$$B \cdot R = B_1R_1 + B_2R_2 + B_3R_3 \quad (21)$$

SUBROUTINE STMPR

PURPOSE: TO PRINT OUT THE TRANSPOSES OF THE STATE TRANSITION
MATRIX PARTITIONS PHI, TXXS, TXW, AND TXU OVER AN
ARBITRARY INTERVAL OF TIME.

CALLING SEQUENCE: CALL STMPR(TRTM1,TRTM2)

ARGUMENT: TRTM1 I TIME AT BEGINNING OF INTERVAL OVER WHICH
STATE TRANSITION MATRIX PARTITIONS HAVE
BEEN COMPUTED

TRTM2 I TIME AT END OF INTERVAL OVER WHICH STATE
TRANSITION MATRIX PARTITIONS HAVE BEEN
COMPUTED

SUBROUTINES SUPPORTED: PRINT4 SETEVS GUISIM GUISS PRESIM
PRINT3 SETEVN GUIDM GUID PRED
PROBE PROBES

COMMON USED: NDIM1 NDIM2 PHI TXU TXXS
XLAB XSL XU
NDIM4 TXW

SUBROUTINE SUB1

PURPOSE: TO COMPUTE POSITION AND VELOCITY MAGNITUDES.

CALLING SEQUENCE: CALL SUB1(X,XE,XP)

ARGUMENT: X I INERTIAL POSITION/VELOCITY OF THE VEHICLE
XE I EARTH-S POSITION/VELOCITY
XP I POSITION/VELOCITY OF THE TARGET PLANET

SUBROUTINES SUPPORTED: PRINT4

LOCAL SYMBOLS: RX MAGNITUDE OF INERTIAL POSITION VECTOR
RY MAGNITUDE OF GEOCENTRIC POSITION VECTOR
RZ MAGNITUDE OF PLANETOCENTRIC POSITION VECTOR
VX MAGNITUDE OF INERTIAL VELOCITY VECTOR
VY MAGNITUDE OF GEOCENTRIC VELOCITY VECTOR
VZ MAGNITUDE OF PLANETOCENTRIC VELOCITY VECTOR
Y GEOCENTRIC POSITION/VELOCITY OF THE VEHICLE
Z PLANETOCENTRIC POSITION/VELOCITY OF THE VEHICLE

SUBROUTINE SUBSOL

PURPOSE: TO COMPUTE TRANSFORMATION MATRIX FROM ECLIPTIC
TO SUBSOLAR ORBITAL COORDINATES

ARGUMENT: D I JULIAN DATE EPOCH 1900

EQSS 0 TRANSFORMATION FROM PLANETOCENTRIC
ECLIPTIC TO SUBSOLAR ORBITAL

NP I INDEX OF PLANET

SUBROUTINES SUPPORTED: SOIPS TARGET TPRTRG

SUBROUTINES REQUIRED: EPHEM ORB USCALE UXV

LOCAL SYMBOLS: G2 MAGNITUDE OF PLANETOCENTRIC VECTOR TO SUN

EXS PLANETOCENTRIC ECLIPTIC VECTOR TO SUN

EYS CROSS PRODUCT OF EZS AND EXS

EZS PLANETOCENTRIC ECLIPTIC UNIT VECTOR
IN DIRECTION OF ANGULAR MOMENTUM OF PLANET

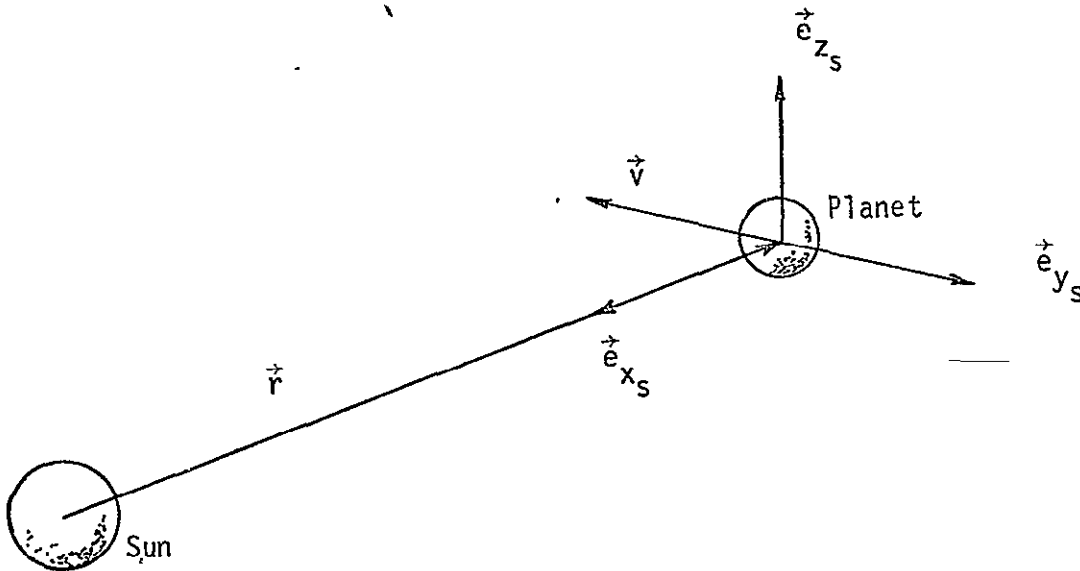
ONE CONSTANT = 1.

COMMON COMPUTED: NO

COMMON USED: XP

SUBSØL Analysis

Subroutine SUBSØL computes the transformation from planetocentric ecliptic coordinates to subsolar planet orbital plane coordinates for an arbitrary planet. The subsolar planet orbital plane coordinate system is defined as the planetocentric system whose x-axis points directly at the sun, whose z-axis is normal to the planet's orbital plane, and whose y-axis is normal to the xz-plane and lies in the planet's orbital plane. In the figure below \vec{r} and \vec{v} denote the position and velocity vectors, respectively, of the planet relative to the sun. Unit vectors \vec{e}_x , \vec{e}_y , and \vec{e}_z are aligned with the axes of the subsolar planet orbital plane system.



These unit vectors are defined as

$$\vec{e}_{x_s} = -\frac{\vec{r}}{r}$$

$$\vec{e}_{y_s} = \vec{e}_{z_s} \times \vec{e}_{x_s}$$

$$\vec{e}_{z_s} = \frac{\vec{r} \times \vec{v}}{|\vec{r} \times \vec{v}|}$$

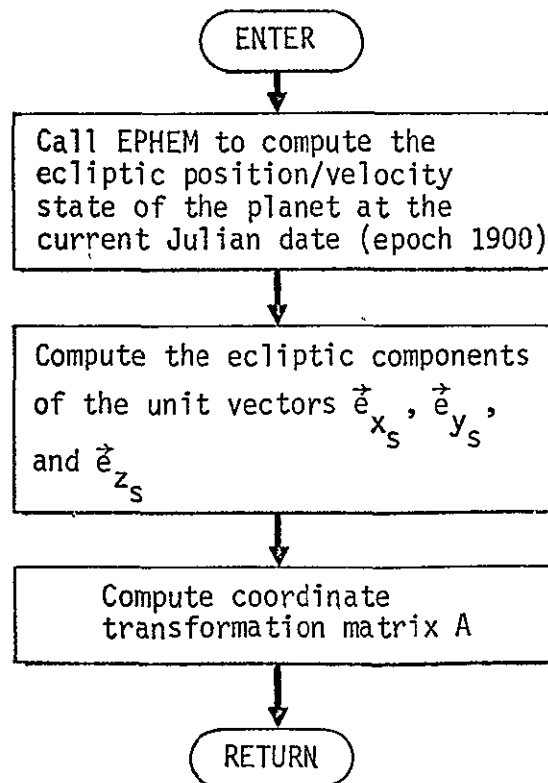
If these unit vectors are referred to the ecliptic coordinate system, the coordinate transformation A from planetocentric ecliptic to subsolar planet orbital plane coordinates is given by

$$A = \begin{bmatrix} \vec{e}_{x_s}^T \\ - - - \\ \vec{e}_{y_s}^T \\ - - - \\ \vec{e}_{z_s}^T \end{bmatrix}$$

Thus

$$\vec{x}_{\text{subsolar}} = A \vec{x}_{\text{ecliptic}}$$

SUBSØL Flow Chart



SUBROUTINE TARGET

PURPOSE: TO PERFORM EXECUTIVE FUNCTIONS OF THE TARGETING MODE AS CALLING REQUIRED SUBROUTINES TO READ THE INPUT DATA, COMPUTING THE ZERO ITERATE IF NECESSARY AND PERFORMING THE ACTUAL TARGETING THROUGH THE PROGRESSIVE STAGES USED BY STEAP.

CALLING SEQUENCE: CALL TARGET

SUBROUTINES SUPPORTED: GIDANS

SUBROUTINES REQUIRED: TAROPT TARMAX DESENT PECEQ VMP

LOCAL SYMBOLS:	ABV	INTERMEDIATE VARIABLE USED TO LIMIT EACH DELTAV COMPONENT CHANGE
	ACC	VECTOR OF ACCURACY LEVELS FOR THE CURRENT TARGETING EVENT
	ACK	ACTUAL ACCURACY USED BY SUBROUTINE VMP
	AER	ABSOLUTE VALUES OF DIFFERENCES BETWEEN DESIRED AND NOMINAL END CONDITIONS
	CERROR	CURRENT SUM OF WEIGHTED DIFFERENCES OF DESIRED AUXILIARY AND NOMINAL AUXILIARY END CONDITIONS
	DEV	DIFFERENCES (ERRORS) BETWEEN AUXILIARY END CONDITIONS (DESIRED AND NOMINAL)
	ISP2	INDICATOR USED BY SUBROUTINE VMP =1 STOP AT SPHERE-OF-INFLUENCE =0 DO NO STOP AT SPHERE-OF-INFLUENCE
	ITBAD	BAQ STEP COUNTER
	ITER	ITERATION COUNTER
	ITOL	CONVERGENCE INDICATOR =1 CASE CONVERGED =0 CASE DID NOT CONVERGE
	IT	INDICATOR USED TO LOCATE DESIRED TIME VALUE FOR OUTER TARGETING
	I	INDEX
	J	INDEX
	LOWHI	INDICATOR USED TO CALCULATE THE PHASE 2

TARGETING MATRIX

NOMORE INDICATOR USED TO LIMIT OUTER TARGETING
 =0 OUTER TARGETING HAS NOT BEEN PERFORMED
 =1 OUTER TARGETING HAS ALREADY BEEN PERFORMED

OSPH ORIGINAL SPHER OF INFLUENCE OF THE TARGET PLANET

PERROR PREVIOUS VALUE OF CERROR

REDUC INTERMEDIATE VARIABLE USED IN BAD STEP REDUCTION

RIS LOCAL VECTOR USED TO SAVE AND RESTORE THE RIN VECTOR

RR INTERMEDIATE VARIABLE FOR OUTER TARGETING

RSF FINAL SPACECRAFT STATE RETURNED BY VMP

SSOI INTERMEDIATE VARIABLE FOR OUTER TARGETING

STOL INTERMEDIATE VARIABLE FOR OUTER TARGETING

TMOF INTERMEDIATE VARIABLE FOR OUTER TARGETING

TVH PHASE 1 TARGETED VELOCITY AT HIGHEST ACCURACY

TVL PHASE 1 TARGETED VELOCITY AT LOSEST ACCURACY

VV INTERMEDIATE VARIABLE FOR OUTER TARGETING

XTIME CURRENT DT TIME USED TO CALCULATE EQECP FOR TARGET PLANET

COMMON COMPUTED/USED:

CTQL	DAUX	DELTAV	DTAR	IBAD
IBAST	IPHASE	ISPH	ISTART	ISTOP
ITARM	KEYTAR	LEVELS	LEV	MATX
MAXBAD	NITS	NOPAR	NOPHAS	NOSOI
PHI	RIN	SPHERE		

COMMON COMPUTED:

DELTP	DELV	ICL2	ICL	INCMT
INPR	IPRINT	KAXTAR	KWIT	RRF

COMMON USED:

AAUX	AC	ALNGTH	ATAR	DC
DELTAT	DT	DVMAX	D1	EQECP
FAC	IBADS	KTAR	KUR	LVLS
HAT	MAXB	NOIT	NPAR	NTP
ONE	RC	SPHFAC	TAR	TM
TOL	TRTM	TWO	ZERO	

TARGET Analysis

TARGET is responsible for the control of any targeting (nonlinear guidance) event. The targeting is done either by the Newton-Raphson technique or by a steepest descent-conjugate gradient algorithm, the method being specified by the user. In either case numerical differencing is used to compute the required sensitivities.

I. Preliminaries

The current inertial state of the spacecraft upon entering TARGET is first saved (RIS=RIN) along with the original SOI radius (OSPH=SPHERE) since both variables may be changed during the course of the targeting. Before exiting from TARGET these values are restored.

The index of the current event KUR has been computed by TRJTRY. This enables the specific targeting parameters for the current event to be set:

Parameter	Definition
METHOD	Triggers Newton Raphson (=0) or Steepest Descent ($\neq 0$) technique
MATX	Determines whether Newton-Raphson matrix is computed always (=2) or only at low level (=1)
IBAST	Determines whether bad step checks are made never (=1), high level only (=2) or always (=3)
LEVELS	Number of integration accuracy levels to be used
NOPAR	Number of target parameters to be used
ACC	Actual accuracy levels used in targeting

The following flags are then initialized to zero

Flag	Definition
ITDS	Counter for steepest descent iterations
LOWHI	Flag indicating whether first phase complete (=1) or not (=0)
NOMORE	Flag indicating whether outer targeting has been done (=1) or not (=0)

The target time is computed and using that time the transformation matrix ϕ_{ECEQ} from ecliptic to target planet equatorial coordinates is calculated (PECEQ).

II. Phase Preparations

TARGET performs the targeting in one phase unless targeting to TCA (time of closest approach). In that case the trajectory is targeted in two phases: the first phase targets to the target planet SOI (sphere of influence), the second phase to the closest approach conditions. IPHASE is the phase counter, NOPHAS is the number of phases needed.

If all the phases have been completed, the program prepares to exit. If the last iterate satisfied the target tolerances ITOL will have been set to a 1. If it did not, ITOL will be zero and this requires that KWET be set to 1 to terminate the program upon return to the basic cycle.

If the last phase has not yet been completed TAROPT is now called with an argument 1 to compute the following phase parameters:

Parameter	Definition
KEYTAR(3)	Vector of codes of target parameters
KAXTAR(3)	Vector of codes of auxiliary parameters
DTAR(3)	Vector of desired values of target parameters
DAUX(3)	Vector of desired values of auxiliary parameters
FAC(3)	Weighting factors for loss function for auxiliary parameters
ISTOP	Flag indicating integration stopping conditions with ISTOP = 1,2,3 indicating fixed final time, SOI, or CA encounter

The target parameters are the parameters actually desired, the auxiliary parameters are the parameters used to do the targeting. The target and auxiliary parameters are identical except when i_{CA} and r_{CA} are targets.

In that case the corresponding auxiliary parameters are B-T and B-R which are much more linear variables. The codes of the target and auxiliary parameters are as follows:

Code	1	2	3	4	5	6	7	8	9	10	11	12
Parameter	TRF*	TSI	TCS	TCA	BDT	BDR	RCA	INC	SMA	XRF	YRF	ZRF

* not currently available

III. Level Preparations

Within any phase TARGET operates through a series of integration accuracy levels prescribed by the user. After completing each level TARGET checks to see if the maximum number of levels LEVELS has been exceeded. If it has the program cycles to the beginning of the "phase loop" to go to the next phase. If the current level LEV is less than LEVELS the following computations are made.

The flag ITARM controls whether the previous targeting matrix is to be used (=1) or whether the matrix is to be recomputed (=2) during the current level. ITARM is set according to the current values of MATX, ISTART, and LEV.

The flag IBAD controls the bad step logic. If IBAD=1 no bad step check will be made during the current level; if IBAD=2 the bad step check will be in effect. TARGET sets IBAD according to the values of IBAST and LEV.

The flags ITOL, ITER, ITBAD are set to 0 to begin the iterations. The allowable iterations NITS and bad iterations MAXBAD are also set at this time.

IV. Iterate Calculations

Within each level the program makes one or more iterations. After each iteration the program updates the iteration counter `ITER`. If the maximum number of iterations for this level `NITS` has been exceeded, the program sets `KWIT` to 1 and prepares for the return from `TARGET`. Otherwise `TARGET` computes the target and auxiliary values corresponding to the current iterate values of state (position and velocity) `RIN`.

The integration parameters are first set. `VMP` is then called to propagate the initial state to the final stopping conditions. Checks are made to insure that the target planet `SOI` was intersected if the stopping conditions were `SOI` or `CA`. If it was not intersected and this is the first iteration, the "outer targeting" phase is entered (see below). If "outer targeting" has already been performed, the bad-step check is entered to reduce the previous correction by `REDUC`.

Otherwise `TAROPT` is called with the argument 2 to compute the desired and actual target (`DTAR`, `ATAR`) and auxiliary (`DAUX`, `AAUX`) parameter values. The absolute error in target values `AER` and the error in auxiliary values `DEV` are then computed.

If the current iterate is the first integration at the low level during the second phase of targeting (`LOWHI=1`) `TARMAX` is now called to compute the phase 2 targeting matrix. Then the state `RIN` is reset to the targeted velocity at the high level `TVH` to prepare for the second phase targeting. The program then returns to the level loop.

Otherwise the program now checks the actual target variables to determine whether they satisfy the input tolerances or not.

V. Tolerances Satisfied

If the tolerances are satisfied, the program first checks to see if the current targeting phase is outer targeting. If it is `TARGET` restores the original target parameters and initiates the normal targeting (see Outer Targeting below).

If the current targeting is already normal targeting, `TARGET` sets `ITOL=1` to indicate the satisfaction of the tolerances. If the problem is a 2-phase and the current level is the highest level in phase 1 targeting, the targeted high level velocity `TVH=RIN` is saved, `LOWHI` is set to 1 and the targeted low level velocity is recalled `RIN=TVL` for the construct of the phase 2 targeting matrix. Then the level loop is reentered.

VI. Bad Step Reduction

If the target parameter values of any iterate are not within the acceptable tolerances `TARGET` now assigns a scalar error ϵ to the iterate using the weighting factors \vec{W}

$$\epsilon = \vec{W} \cdot \Delta \vec{T}$$

If the bad-step check is to be made on this iterate the current error ϵ is compared to the previous error ϵ_p . If $\epsilon > \epsilon_p$ and the maximum number of bad steps has not been exceeded, the previous correction $\Delta \vec{v}$ is reduced by REDUC (usually 1/4). The initial state RIN is adjusted by this and the iterate loop is reentered. If the maximum number of bad steps has been made, KWIT is set to 1 and the preparations for return are made.

VII. Generation of Next Iterate

The correction $\Delta \vec{v}$ to any iterate may be computed from either of two techniques selected by the flag METHOD. If METHOD \neq 0, subroutine DESCENT is called for the computation of the $\Delta \vec{v}$ by a steepest descent algorithm. The numerical value of METHOD determines the number n of conjugate gradient steps between each straight gradient step where $n = \text{METHOD} - 1$. Thus if METHOD=1, every step is in the direction of the gradient. But if METHOD=5, four steps are taken following the conjugate gradient direction before rectification by the gradient direction.

If METHOD=0, the Newton-Raphson correction is used. If ITARM=0, TARMAX is called for the computation of the targeting matrix ϕ by numerical differencing. If any of the integrations made in constructing that matrix satisfy the tolerances in τ , the flag IEND is set to 1 before returning to TARGET. Thus a check must be made on IEND. If ITARM=1 the previous targeting matrix is used. The correction is then given by

$$\Delta \vec{v} = \phi \cdot \Delta \vec{\alpha}$$

where $\Delta \vec{\alpha}$ are the deviations in the iterate auxiliary values. The $\Delta \vec{v}$ is checked to insure that the maximum step size DVMAX is not violated: if it is, the $\Delta \vec{v}$ is reduced proportionately to satisfy it. The next iterate is then set to

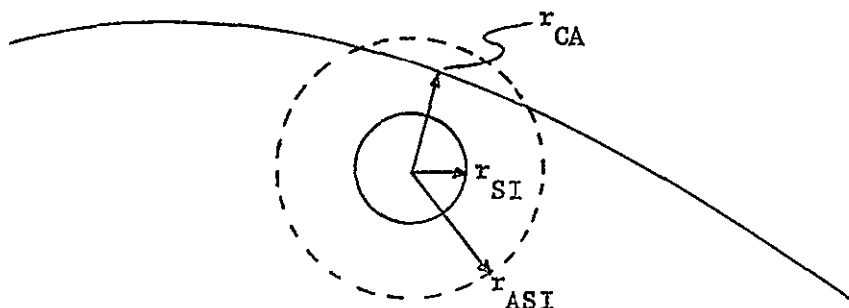
$$(\vec{r}, \vec{v}) = (\vec{r}, \vec{v} + \Delta \vec{v})$$

and the return is made to the iterate loop.

VIII. Outer Targeting

Occasionally the zero iterate initial state leads to a trajectory missing the target body SOI. Since all target options except one (targeting to a specified position, i.e., KTAR = 10,11,12) require the trajectory to intersect the target body SOI steps must be taken to correct this.

Let the initial state propagated forward lead to a trajectory with a closest approach to the target body of r_{CA} with $r_{CA} > r_{SI}$ where r_{SI} is the radius of the SOI.



Until the initial trajectory intersects the SOI the usual targeting can not be done. Therefore an "artificial" SOI is introduced having a radius of

$$r_{ASI} = 1.2 \times r_{CA}$$

The initial trajectory obviously intersects the artificial SOI and hence may be targeted to conditions on the ASOI. If the target conditions are established as $B \cdot T_A = B \cdot R_A = 0$, when this artificial targeting is completed, the refined trajectory will be headed straight for the target body when it hits the ASOI. Thus the refined trajectory should automatically hit the normal SOI when propagated past the ASOI. To insure that the time of intersection with the normal SOI is consistent with the target time, an artificial target time is also used. Let the speed of the spacecraft with respect to the target body at r_{CA} be v_{CA} . Make the approximation that this speed will be roughly the same for the refined trajectory. Then the time that the spacecraft should intersect the ASOI is

$$t_{ASI} = t_{CA} - \frac{r_{ASI}}{v_{CA}}$$

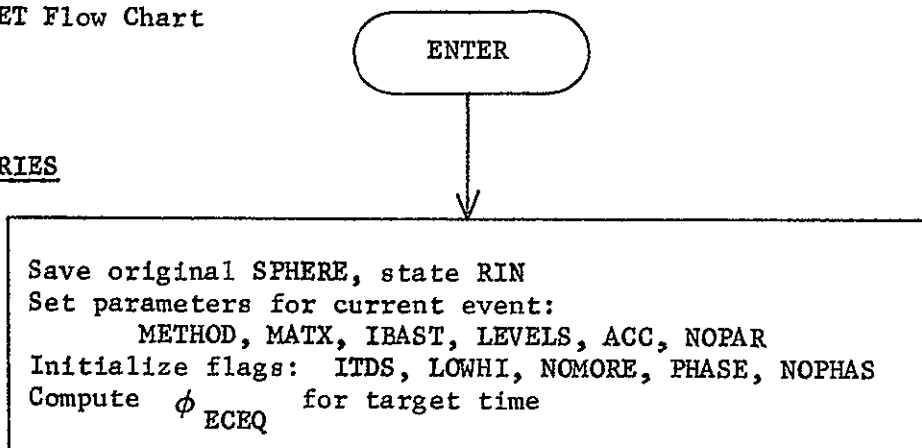
$$\text{or} \quad t_{ASI} = t_{SI} - \frac{r_{ASI} - r_{SI}}{v_{CA}}$$

where the first formula should be used if the target time is t_{CA} or t_{CS} and the second formula is used for t_{SI} .

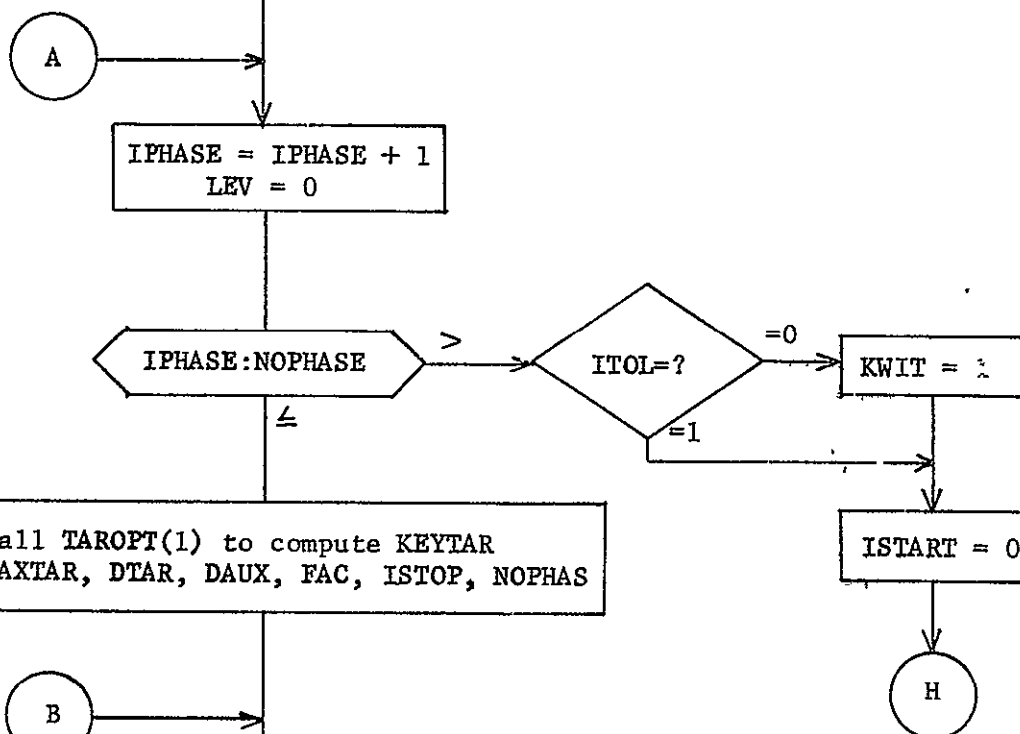
Thus when a trajectory is found which misses the normal SOI, the closest approach state r_{CA}, v_{CA} is recorded. The normal SOI radius is stored, and the artificial SOI radius given above is used in its place. Target parameters of $B \cdot T_A, B \cdot R_A$, and t_{ASI} are then set up as the targets. When targeting of this artificial problem is complete, the resulting trajectory will intersect the normal SOI and the original problem may be solved.

TARGET Flow Chart

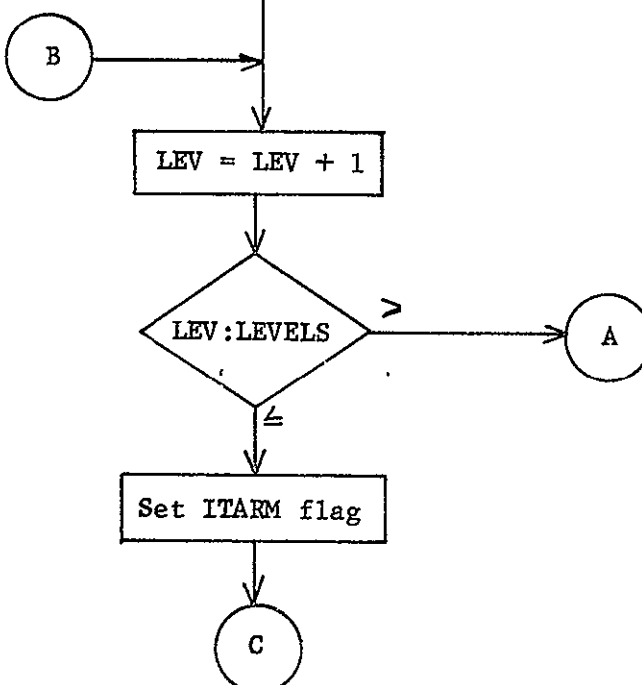
PRELIMINARIES

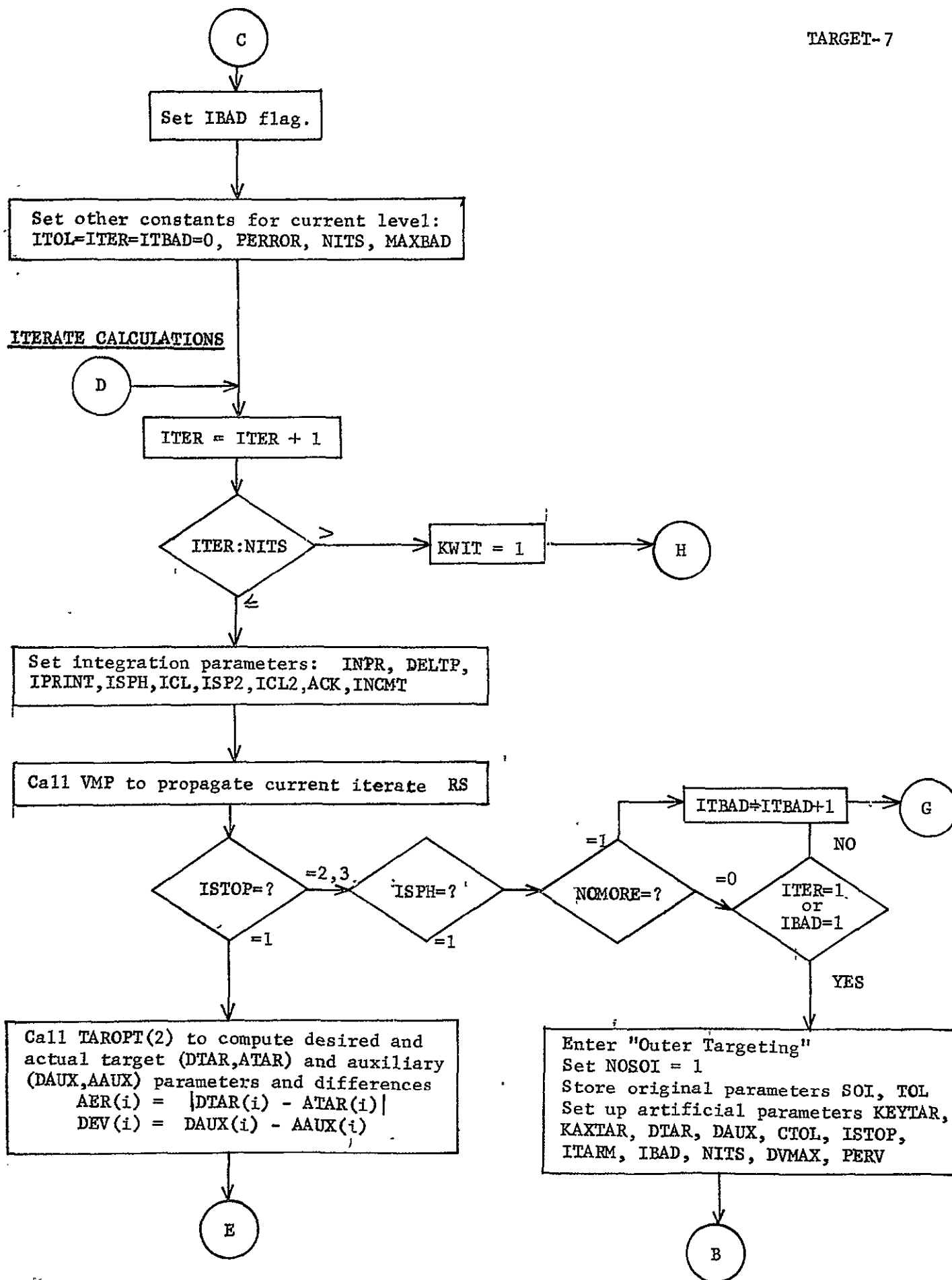


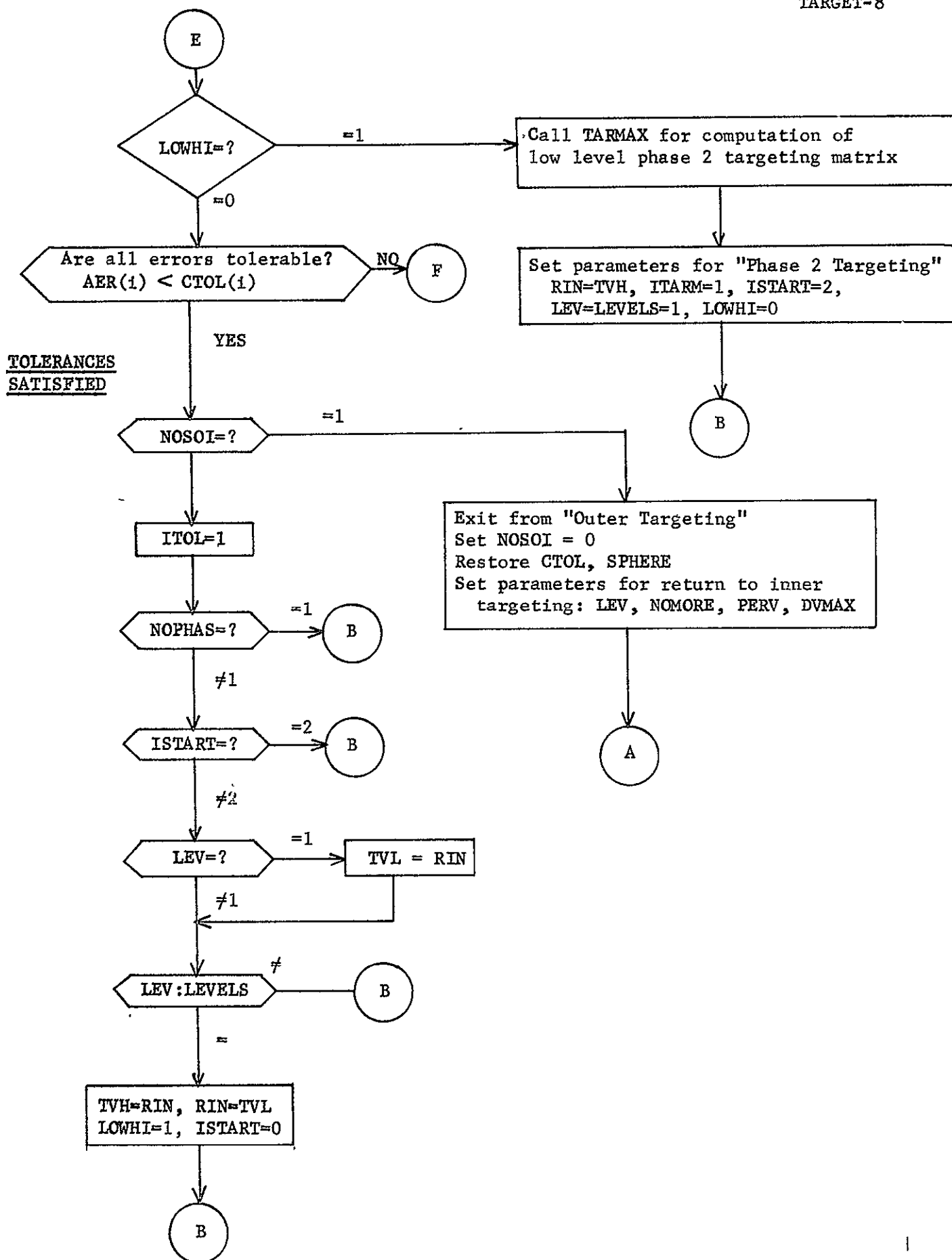
PHASE PREPARATIONS



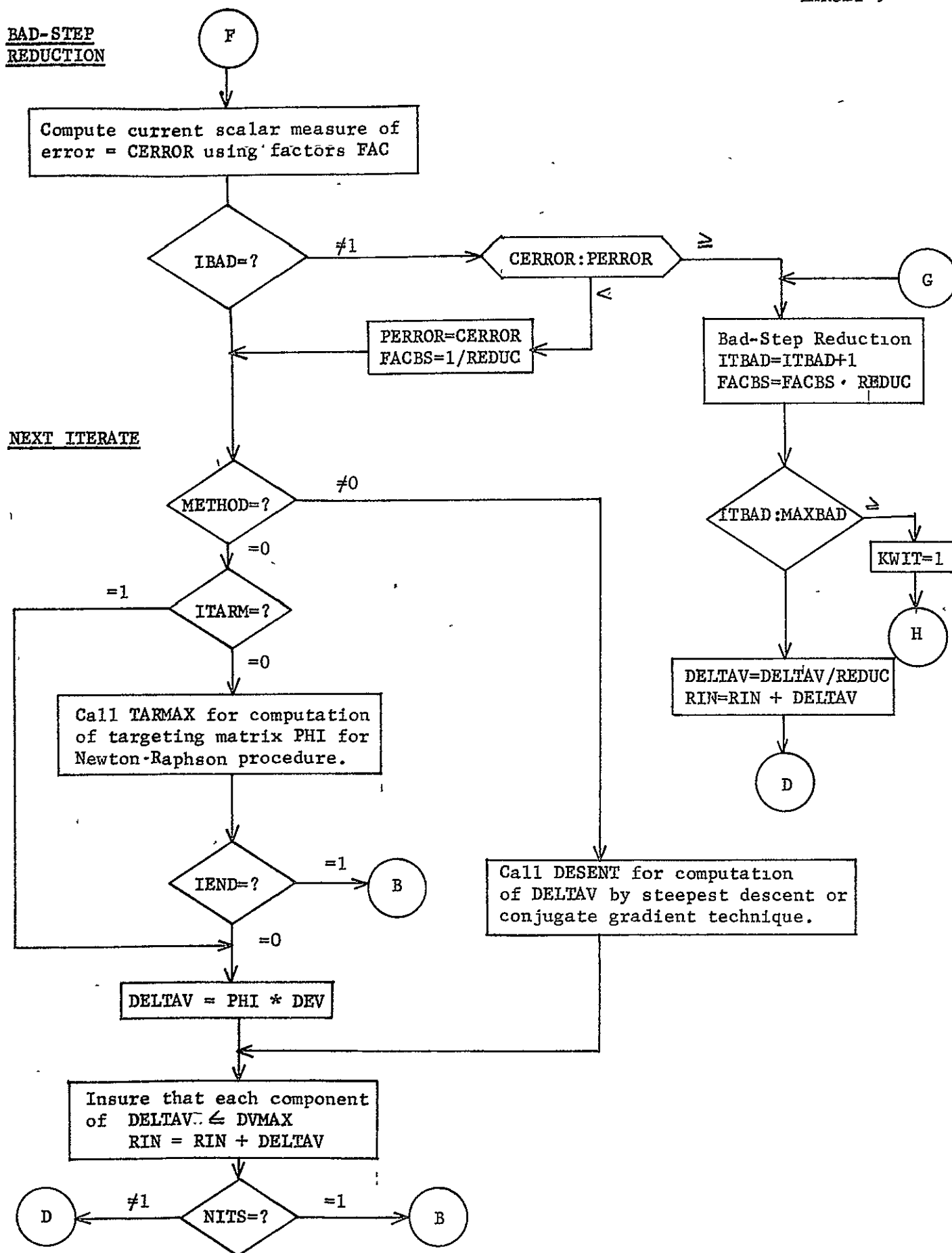
LEVEL PREPARATIONS

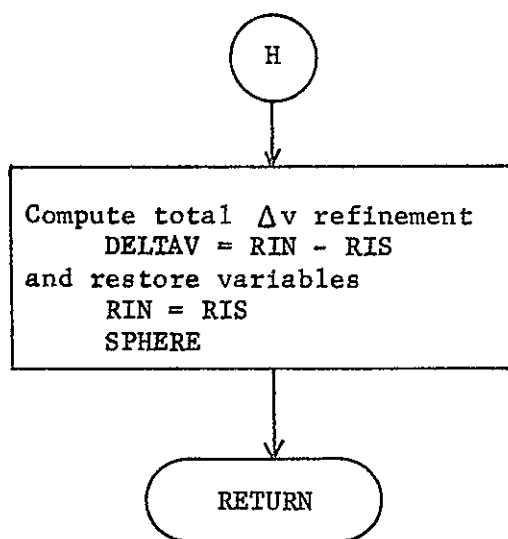






BAD-STEP
REDUCTION



PREPARATIONS FOR RETURN

SUBROUTINE TARMAX

PURPOSE: TO CALCULATE A TARGET MATRIX FROM NOMINAL INJECTION CONDITIONS, AND A PERTURBATION FACTOR BDELV FOR A GIVEN ACCURACY LEVEL.

CALLING SEQUENCE: CALL TARMAX

SUBROUTINES SUPPORTED: TARGET

SUBROUTINES REQUIRED: MATIN TAROPT VMP

LOCAL SYMBOLS: ACK ACCURACY USED TO GENERATE THE TARGET MATRIX

AER DIFFERENCES BETWEEN DESIRED AND ACTUAL END CONDITIONS

AUXN NOMINAL AUXILIARY END CONDITIONS

CHI STATE TRANSITION MATRIX RELATING PERTURBATIONS IN THE RIN VECTOR TO CHANGES IN AUXN

DVEE VECTOR OF VELOCITY COMPONENT PERTURBATIONS

ISP2 INDICATOR USED BY SUBROUTINE VMP
=0 DO NOT STOP AT SPHERE OF INFLUENCE
=1 STOP AT SPHERE OF INFLUENCE

I INDEX

J INDEX

KOMP INDEX

PSI TARGET MATRIX FOR 2 X 2 EASE, STORED INTO PHI

RSF FINAL SPACECRAFT STATE RETURNED BY VMP

COMMON COMPUTED/USED: ISPH PHI RIN TRTH

COMMON COMPUTED: ECL2 ICL INCHT

COMMON USED: AAUX AC ATAR CTOL DAUX
DELTAT DELTAV DTAR D1 ISTOP
KUR LEV LVLS NOPAR PERV
ZERO

TARMAX Analysis

TARMAX computes the targeting matrix used by TARGET for Newton-Raphson refinements. The targeting matrix is computed by numerical differencing.

Let the current iterate initial state be denoted \vec{r}, \vec{v} . Let the auxiliary parameters corresponding to this state be $\vec{\alpha}$. Let the perturbation size for the sensitivities be Δv .

The k-th column of the sensitivity matrix is computed as follows. Perturb the k-th component of velocity by Δv :

$$\vec{v}_p = \vec{v} + \Delta v \begin{bmatrix} \delta_{1K} & \delta_{2K} & \delta_{3K} \end{bmatrix}^T \quad (1)$$

Propagate the perturbed initial state (\vec{r}, \vec{v}_p) to the final stopping conditions. Let the auxiliary parameters of that trajectory be denoted α_p . The k-th column of the sensitivity matrix x is then given by

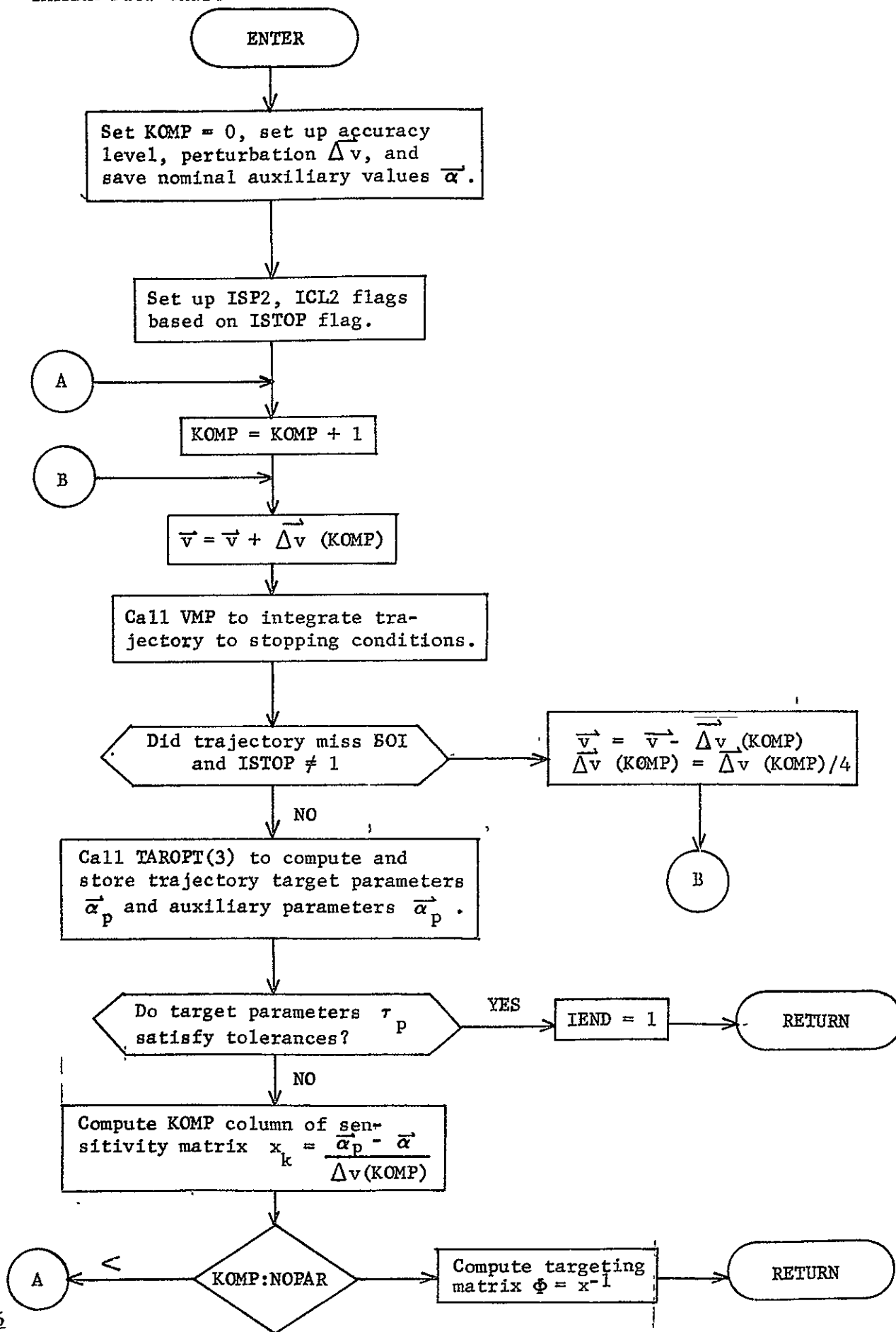
$$x_k = \frac{\vec{\alpha}_p - \vec{\alpha}}{\Delta v} \quad (2)$$

Having computed all the columns of x , the targeting matrix is then given by the inverse of x :

$$\Phi = x^{-1} \quad (3)$$

The targeting matrix then has the property that to obtain a change $\Delta \alpha$ in the nominal auxiliary parameters, the velocity should be changed by the amount

$$\vec{\Delta v} = \Phi \cdot \vec{\Delta \alpha} \quad (4)$$



SUBROUTINE TAROPT

PURPOSE: TO COMPUTE THE DESIRED AND ACHIEVED TARGET PARAMETER
VALUES FOR ALL THE TARGETING SUBROUTINES.

CALLING SEQUENCE: CALL TAROPT(ITARO)

ARGUMENT: ITARO I OPTION FLAG
 =1 SET UP TARGETING PARAMETERS FOR TARGET
 KEYS
 =2 COMPUTE ACTUAL VALUES OF PARAMETERS
 =3 COMPUTE ACTUAL AND DESIRED VALUES OF
 PARAMETERS

SUBROUTINES SUPPORTED: TARGET TARMAX DESENT

SUBROUTINES REQUIRED: CAREL CPWMS IMPACT

LOCAL SYMBOLS: ACK CURRENT ACCURACY BEING USED

A SEMI-MAJOR AXIS OF THE TARGET
PLANETOCENTRIC CONIC

CPT TOTAL COMPUTER TIME USED (SECS)

DBDR DESIRED VALUE OF B DOT R

DBDT DESIRED VALUE OF B DOT T

DINC DESIRED VALUE OF INCLINATION

DRCA DESIRED VALUE OF RCA

E ECCENTRICITY OF THE TARGET PLANETOCENTRIC
CONIC

IAUX INDICATOR FOR AUXILIARY END CONDITIONS
 =0 TARGET TO ACTUAL END CONDITIONS
 =1 TARGET TO AUXILIARY END CONDITIONS

IINC LOCATES DESIRED INCLINATION IN THE DTAR
ARRAY

IRCA LOCATES DESIRED RCA IN THE DTAR ARRAY

I INDEX

KEY LOCAL VARIABLE USED TO COMPLETE
INFORMATION IN THE KAXTAR AND KEYTAR ARRAY

PP DUMMY VARIABLE FOR CALL TO CAREL

QQ DUMMY VARIABLE FOR CALL TO CAREL

RM *du* MAGNITUDE OF SPACECRAFT USED TO COMPUTE
 SEMI-MAJOR AXIS
 TA DUMMY ARGUMENT FOR CALL TO CAREL
 TDBR DUMMY ARGUMENT FOR CALL TO IMPACT
 TBDT DUMMY ARGUMENT FOR CALL TO IMPACT
 TFP TIME OF FLIGHT FROM PERIAPSIS ON THE
 TARGET PLANETOCENTRIC CONIC
 TIMC INTERMEDIATE VARIABLE TO COMPUTE CPT
 TSICA DUMMY VARIABLE FOR CALL TO IMPACT
 VX INTERMEDIATE VARIABLE USED TO CALCULATE
 SEMI-MAJOR AXIS FOR OPTION 9
 WW DUMMY VARIABLE FOR CALL TO CAREL
 W DUMMY VARIABLE FOR CALL TO CAREL
 XI DUMMY VARIABLE FOR CALL TO CAREL
 XN DUMMY VARIABLE FOR CALL TO CAREL

COMMON COMPUTED/USED: AAUX ATAR DAUX DELTAT DTAR
 ISTOP KAXTAR KEYTAR NOPAR NOPHAS
 RCA

COMMON COMPUTED: CTOL FAC

COMMON USED: AC BDR BOT CAINC DC
 DG DSI DT EQECP IBAD
 ICL2 INCHT IPHASE KTAR KUR
 LEV NOSOI NPAR ONE RC
 RIN RRF RSI TAR TIMS
 TMU TM TOL TWO VSI

TAROPT Analysis

TAROPT is responsible for computing the desired and achieved target parameter values for all the targeting subroutines. To add any new target parameters, TAROPT is the only subroutine that must be modified.

The key variables used by TAROPT and their definitions are:

Variable	-	Definition;
KTAR(6,10)	-	Codes of target parameters of all targeting event
TAR(6,10)	-	Desired values of target parameters of all targeting events;
KEYTAR(3)	-	Codes of target parameters of current event;
DTAR(3)	-	Desired values of target parameters of current event;
ATAR(3)	-	Actual values of target parameters of current iterate;
KAXTAR(3)	-	Codes of auxiliary parameters of current iterate;
DAUX(3)	-	Desired values of auxiliary parameters on current iterate;
AAUX(3)	-	Actual values of auxiliary parameters of current iterate.

The available target parameters and their codes and definitions are tabulated.

Code	Parameter	Definition
1	t_{PS}	Time at probe sphere impact (n-body integration to sphere of influence (SOI), conic propagation to probe-sphere)
2	t_{SI}	Time at SOI of target body (n-body integration to SOI)
3	t_{CS}	Time at CA (n-body integration to SOI, conic propagation to CA)
4	t_{CA}	Time at CA (n-body integration to CA)
5	$B \cdot T$	Impact parameter $B \cdot T$
6	$B \cdot R$	Impact parameter $B \cdot R$
7	i	Inclination to target planet equator
8	r_{CA}	Radius of closest approach to target body
9	a_{SI}	Semimajor axis of conic with respect to target body
10	x_f	X-component of final state (inertial ecliptic system)
11	y_f	Y-component of final state
12	z_f	Z-component of final state
13	δ_I	Declination of probe target point in planetocentric probe-sphere coordinate system specified by IPCS
14	α_I	Right ascension of probe target point in planetocentric probe-sphere coordinate system specified by IPCS
15	t_{PR}	Time at probe-sphere impact (n-body integration to probe sphere)

The term target parameter refers to a variable with a final value that conforms to a desired value. The term auxiliary parameter refers to a variable used to compute the progressive corrections. The target parameters and auxiliary parameters are identical unless the target parameters include either of the pairs i and r_{CA} or α_I and δ_I . In these cases the more linear parameters $B \cdot T$ and $B \cdot R$ are used as auxiliary targets in place of the actual ones. The desired values of the asymptote pierce-point coordinates, $B_D \cdot T$ and $B_D \cdot R$, as well as the actual values, $B_A \cdot T$ and $B_A \cdot R$, are computed by IMPCT for each successive iterate of the trajectory, based on the desired values of i and r_{CA} or α_I and δ_I .

TAROPT is called under three different options, which are distinguished by an argument ITARØ. The three different options will be discussed in order.

TAROPT(1) is called by TARGET at the beginning of each phase to set up the proper variables for the targeting. The arrays KEYTAR, KAXTAR, DTAR, and DAUX are set to the current event values of KTAR and TAR. If t_{CA} or t_{PR} is a target parameter, the number of phases NØPHAS is set to 2. Then for the first phase of a two-phase problem, the time target variable t_{CA} or t_{PR} is replaced by t_{CS} or t_{PS} , respectively, in the KEYTAR and KAXTAR arrays. Thus in the initial high-speed phase of a high-precision double-phase targeting problem, integration is stopped at the SOI and the trajectory is extrapolated to the target conditions as in a single-phase case. If i and r_{CA} or α_I and δ_I are included in the actual target parameters, the corresponding indices of the KAXTAR array are set for the auxiliary targets $B \cdot T$ and $B \cdot R$. TAROPT then sets up the integration parameters.

There are three propagation stop modes. The first terminates the trajectory after a maximum propagation time interval, Δt , set equal to the nominal difference between the target time t_T and the current guidance event time t_G :

$$\Delta t = t_T - t_G \quad (1)$$

For this stopping condition, ISTØP is set to 1. The second termination mode stops propagation at the SOI (or at impact if the SOI radius is temporarily set to that of the probe sphere). This ISTØP = 2 mode is used when the target time is t_{SI} , t_{CS} , t_{PS} , or t_{PR} .

The final ISTOP = 3 mode terminates the propagation at closest approach. It is used only when the target time is t_{CA} . In either of the last two modes, the maximum propagation time interval is set to 1.1 times the value assigned for the first mode except when one of the target parameters is a_{SI} , the semimajor axis of the planetocentric conic at the target time. In this case Δt is set to the same value as in the first stopping mode. The augmented propagation interval guarantees that the trajectory will be propagated long enough to reach the desired targeting termination.

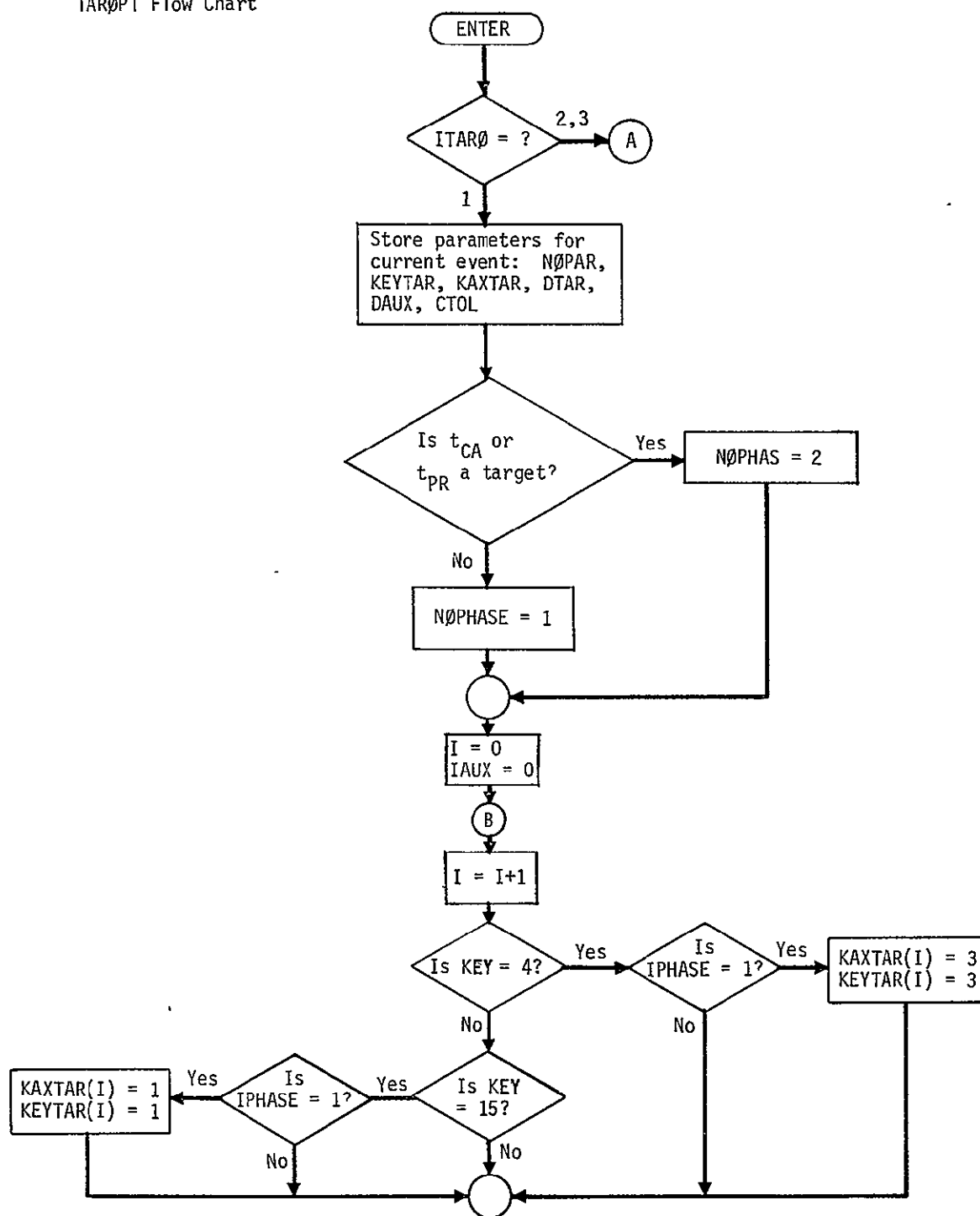
Finally the weighting factors FAC(3) used in computing the scalar loss function are set. Since the loss function is calculated solely from the auxiliary targets, which all have units of either length or time, only the relative weight of time to length need be input. Thus the length factors are set to unity and the time factor to the input value in WGHM.

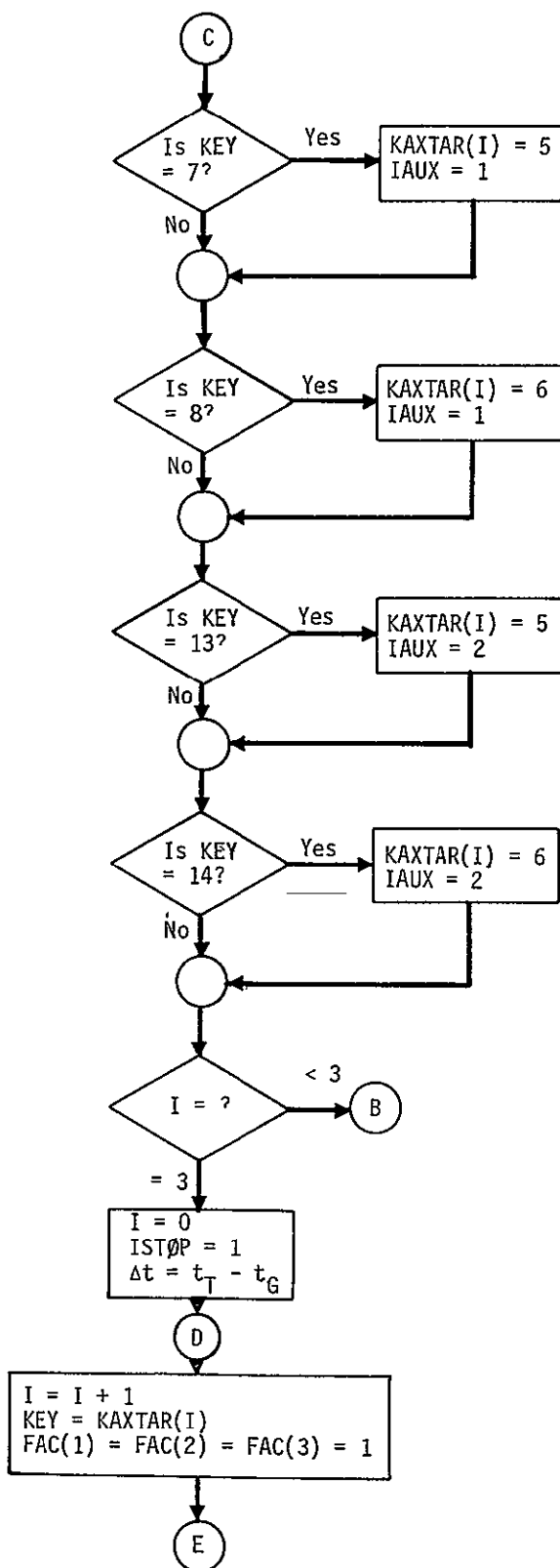
TAROPT(2) is called by TARGET after integrating each iterate to the final stopping conditions. Here TAROPT performs mainly a bookkeeping role. It must fill the proper cells of the ATAR, AAUX, and DAUX arrays with values generally computed by the virtual-mass routines. If auxiliary targeting is required, both the actual and desired values of the B-plane coordinates are computed by calling IMPCT.

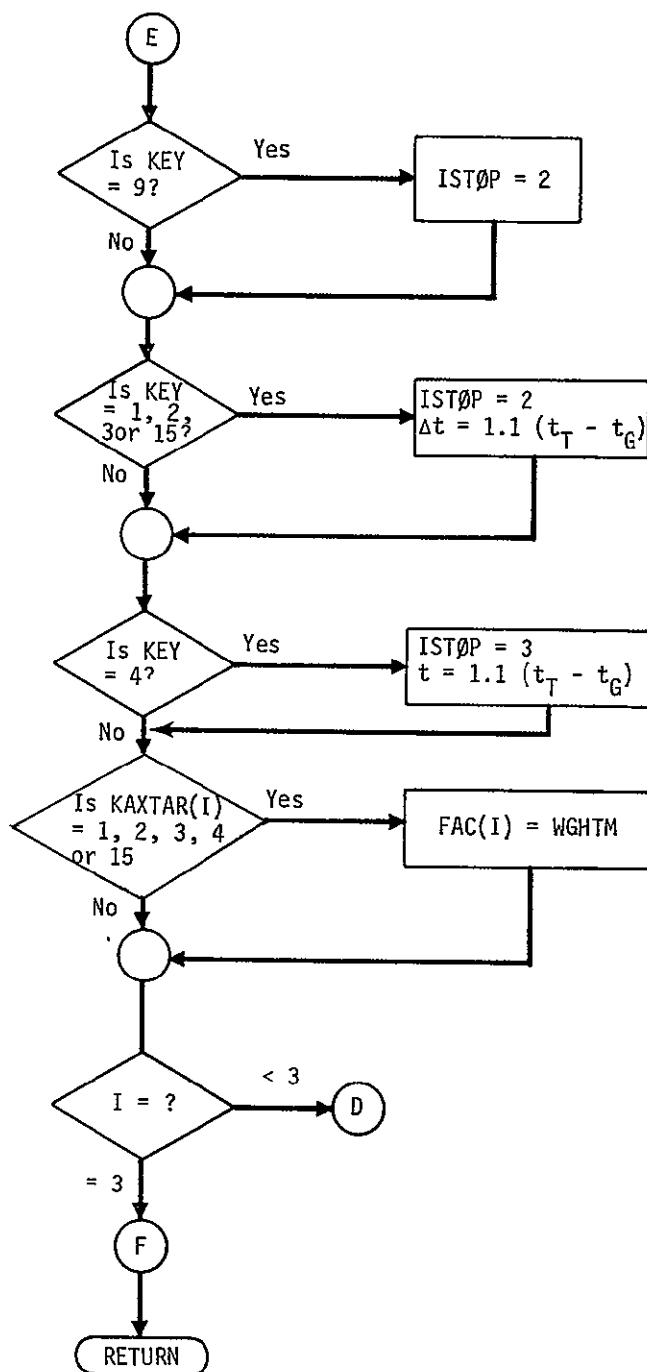
Since TAROPT(3) is called by TARMAX and DESENT after integrating each perturbed trajectory to compute the perturbed values of the auxiliary parameters, the desired values of DAUX need not be computed at this time. If auxiliary targeting is in process, the actual values of the B-plane coordinates, $B_A \cdot T$ and $B_A \cdot R$, are computed by a call to IMPCT. Once again this task is simply a bookkeeping job to store the trajectory data correctly in the ATAR and AAUX cells. TARMAX and DESENT may then operate easily on these arrays to compute the targeting matrix or gradient directions.

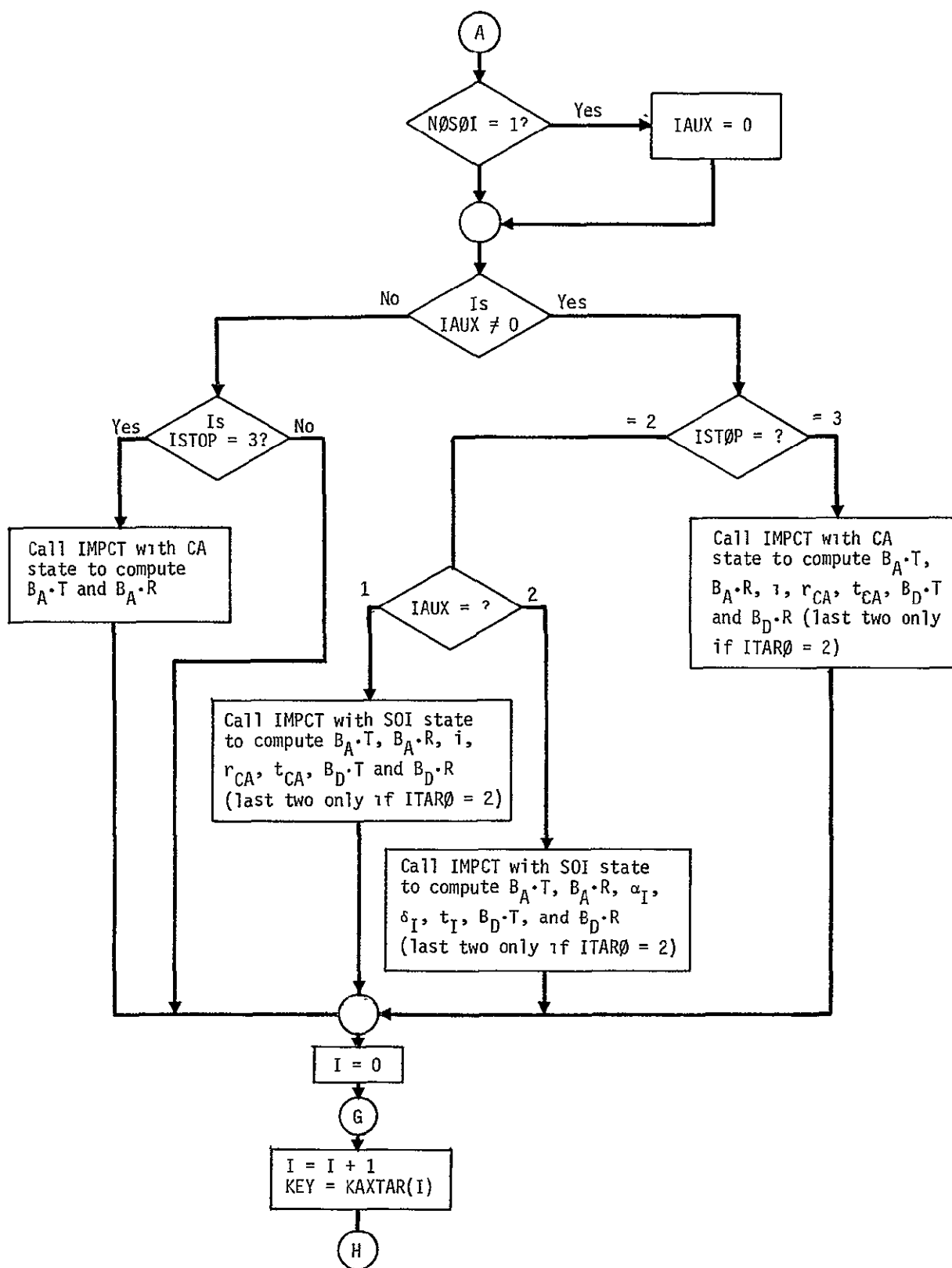
In both calls TAROPT(2) and TAROPT(3), the trajectory data are printed out before exiting from TAROPT.

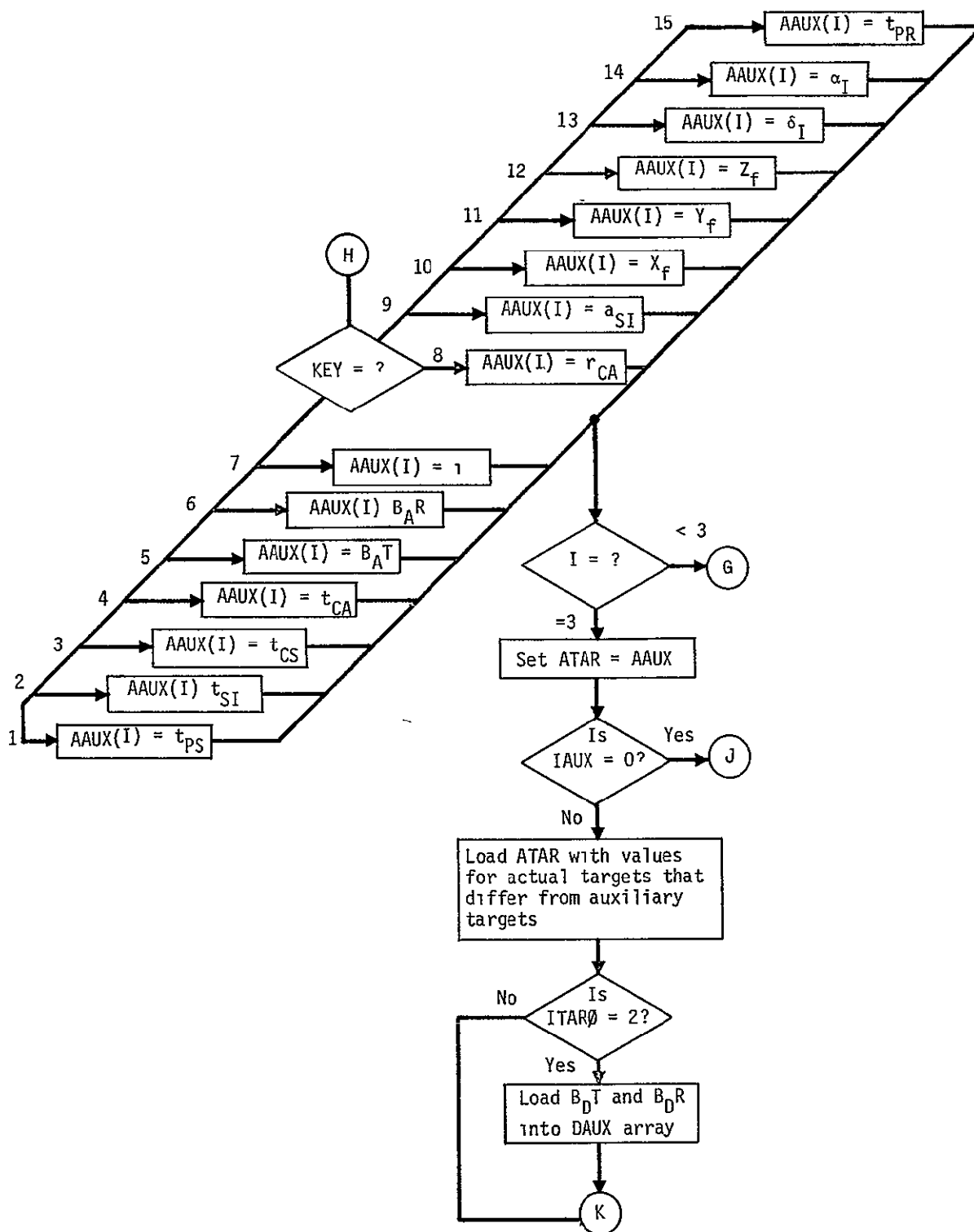
TAROPT Flow Chart

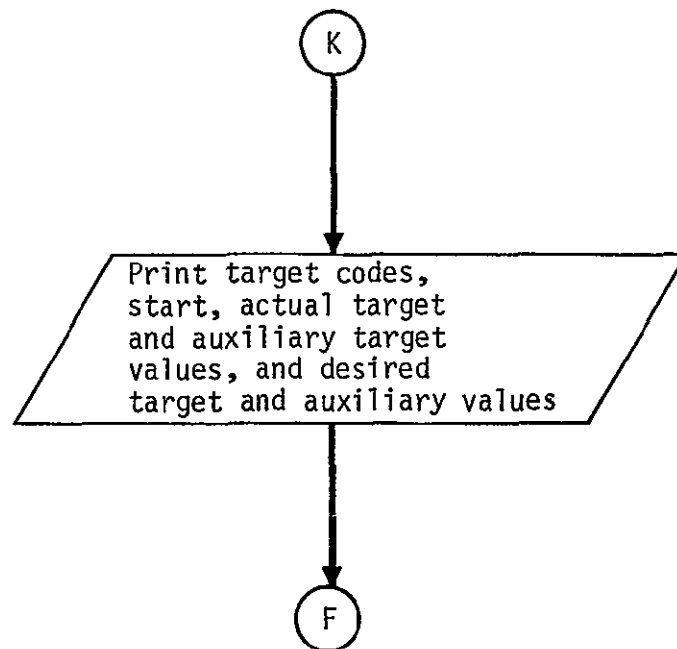












SUBROUTINE TARPRL

PURPOSE: TO COMPUTE THE PARTIAL DERIVATIVES OF THE POSITION COMPONENTS OF A PLANET WITH RESPECT TO EACH OF ITS ORBITAL ELEMENTS.

CALLING SEQUENCE: CALL TARPRL(ICODE,PAR)

ARGUMENT: ICODE I CODE DEFINING ORBITAL ELEMENT OF INTEREST
 PAR 0 VECTOR OF 3 POSITION PARTIALS WITH RESPECT TO THE ORBITAL ELEMENT OF INTEREST

SUBROUTINES SUPPORTED: TRAKS TRAKM

LOCAL SYMBOLS: CBO COSINE OF LONGITUDE OF ASCENDING NODE
 CI COSINE OF ANGLE OF INCLINATION
 CLO COSINE OF ARGUMENT OF PERIAPSIS
 COSNU COSINE OF TRUE ANOMALY
 COSONU COSINE OF THE SUM OF THE ANGLES OF TRUE ANOMALY PLUS THE ARGUMENT OF PERIAPSIS
 DNUDE PLANET DISTANCE TIMES THE PARTIAL OF TRUE ANOMALY WITH RESPECT TO ECCENTRICITY
 DNUDM PARTIAL OF TRUE ANOMALY WITH RESPECT TO MEAN ANOMALY
 DPAR $1./R*(\#R/\#E + (\#R/\#NU)*(\#NU/\#E))$
 WHERE R= PLANET DISTANCE
 NU=TRUE ANOMALY
 E= ECCENTRICITY
 #= PARTIAL OF
 DRDNU PARTIAL R WITH RESPECT TO NU
 E2 SQUARE OF ECCENTRICITY
 IND INDEX USED IN ARRAY STORING ORBITAL ELEMENTS OF PLANETS
 PCOMP SEMI-MAJOR AXIS TIMES THE TERM $(1-E^2)$
 WHERE E=ECCENTRICITY
 R PLANET DISTANCE
 SBO SINE OF LONGITUDE OF THE ASCENDING NODE
 SI SINE OF INCLINATION

SINNU SINE OF TRUE ANOMALY
SINONU SINE OF THE SUM OF THE ANGLES OF TRUE
ANOMALY PLUS THE ARGUMENT OF PERIAPSIS
SLO SINE OF THE ARGUMENT OF PERIAPSIS
XXX SCRATCH CELL

COMMON USED:

ALNGTH ELMNT NTP ONE XP
ZERO

TARFRL Analysis

The position components of a planet are related to its orbital elements a , e , i , Ω , ω , and M through the following set of equations:

$$x = r \left[\cos \Omega \cos(\omega + \nu) - \sin \Omega \sin(\omega + \nu) \cos i \right] \quad (1)$$

$$y = r \left[\sin \Omega \cos(\omega + \nu) + \cos \Omega \sin(\omega + \nu) \cos i \right] \quad (2)$$

$$z = r \sin(\omega + \nu) \sin i \quad (3)$$

$$r = \frac{a(1 - e^2)}{1 + e \cos \nu} \quad (4)$$

$$\tan \frac{\nu}{2} = \sqrt{\frac{1+e}{1-e}} \tan \frac{E}{2} \quad (5)$$

$$M = E - e \sin E \quad (6)$$

We can write equations (1), (2), (3), and (4) symbolically as

$$x_1 = f_1(a, e, i, \Omega, \omega, \nu)$$

and equations (5) and (6) as

$$\nu = \nu(e, M)$$

Then the partials of x_1 with respect to a , e , i , Ω , ω , and M can be evaluated as follows:

$$\frac{\partial x_1}{\partial a} = \frac{\partial f_1}{\partial a} \quad (7)$$

$$\frac{\partial x_1}{\partial e} = \left(\frac{\partial f_1}{\partial e} \right)_{\nu} + \frac{\partial f_1}{\partial \nu} \cdot \frac{\partial \nu}{\partial e} \quad (8)$$

$$\frac{\partial x_1}{\partial i} = \frac{\partial f_1}{\partial i} \quad (9)$$

$$\frac{\partial x_1}{\partial \Omega} = \frac{\partial f_1}{\partial \Omega} \quad (10)$$

$$\frac{\partial x_1}{\partial \omega} = \frac{\gamma f_1}{\partial \omega} \quad (11)$$

$$\frac{\partial x_1}{\partial M} = \frac{\partial f_1}{\partial \nu} \cdot \frac{\partial \nu}{\partial M} \quad (12)$$

Only $\frac{\partial \nu}{\partial e}$ and $\frac{\partial \nu}{\partial M}$ require further consideration before equations (7) through (11) can be used to obtain expressions for the 18 desired partial derivatives.

We obtain $\frac{\partial \nu}{\partial M}$ by first differentiating equation (5) with respect to E and equation (6) with respect to M to obtain

$$\frac{\partial \nu}{\partial E} = \frac{a}{r} \sqrt{1 - e^2}$$

and

$$\frac{\partial E}{\partial M} = \frac{a}{r}$$

Then

$$\frac{\partial \nu}{\partial M} = \frac{\partial \nu}{\partial E} \cdot \frac{\partial E}{\partial M} = \left(\frac{a}{r} \right)^2 \sqrt{1 - e^2} \quad (13)$$

We obtain $\frac{\partial \nu}{\partial e}$ by first differentiating equation (6) with respect to e to obtain

$$\frac{\partial M}{\partial e} = - \frac{\sqrt{1 - e^2} \sin \nu}{1 + e \cos \nu}$$

This result is then combined with equation (13) to yield

$$\frac{\partial \nu}{\partial e} = \frac{\partial \nu}{\partial M} \frac{\partial M}{\partial e} = - \left(\frac{a}{r} \right)^2 \frac{(1 - e^2) \sin \nu}{1 + e \cos \nu} \quad (14)$$

The evaluation of the desired partials can now proceed. The results are summarized below.

a. Partial with respect to a .

$$\frac{\partial x}{\partial a} = \frac{x}{a}$$

$$\frac{\partial y}{\partial a} = \frac{y}{a}$$

$$\frac{\partial z}{\partial a} = \frac{z}{a}$$

b. Partial with respect to e .

$$\frac{\partial x}{\partial e} = \frac{xq}{r} + r \frac{\partial \nu}{\partial e} \left[-\cos \Omega \sin(\omega + \nu) - \sin \Omega \cos(\omega + \nu) \cos i \right]$$

$$\frac{\partial y}{\partial e} = \frac{yq}{r} + r \frac{\partial \nu}{\partial e} \left[-\sin \Omega \sin(\omega + \nu) + \cos \Omega \cos(\omega + \nu) \cos i \right]$$

$$\frac{\partial z}{\partial e} = \frac{zq}{r} + r \frac{\partial \nu}{\partial e} \cos(\omega + \nu) \sin i$$

$$\text{where } q = \frac{r}{ae(1 - e^2)} \left[r - a - ae^2(1 + \sin^2 \nu) \right]$$

c. Partial with respect to i .

$$\frac{\partial x}{\partial i} = r \sin \Omega \sin(\omega + \nu) \sin i$$

$$\frac{\partial y}{\partial i} = -r \cos \Omega \sin(\omega + \nu) \sin i$$

$$\frac{\partial z}{\partial i} = r \sin(\omega + \nu) \cos i$$

d. Partial with respect to Ω .

$$\frac{\partial x}{\partial \Omega} = -y$$

$$\frac{\partial y}{\partial \Omega} = x$$

$$\frac{\partial z}{\partial \Omega} = 0$$

e. Partial derivatives with respect to ω .

$$\frac{\partial x}{\partial \omega} = r \left[-\cos \Omega \sin(\omega + \nu) - \sin \Omega \cos(\omega + \nu) \cos i \right]$$

$$\frac{\partial y}{\partial \omega} = r \left[-\sin \Omega \sin(\omega + \nu) + \cos \Omega \cos(\omega + \nu) \cos i \right]$$

$$\frac{\partial z}{\partial \omega} = r \cos(\omega + \nu) \sin i$$

f. Partial derivatives with respect to M .

$$\frac{\partial x}{\partial M} = \frac{xs}{r} + r \frac{\partial \nu}{\partial M} \left[-\cos \Omega \sin(\omega + \nu) - \sin \Omega \cos(\omega + \nu) \cos i \right]$$

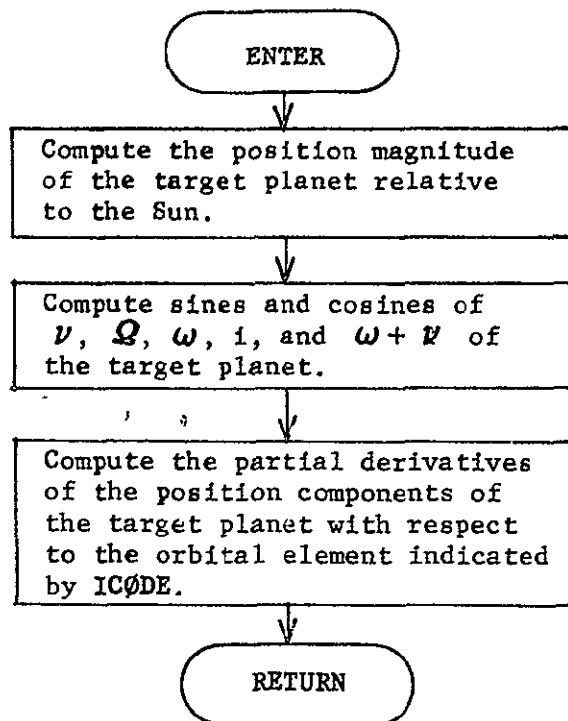
$$\frac{\partial y}{\partial M} = \frac{ys}{r} + r \frac{\partial \nu}{\partial M} \left[-\sin \Omega \sin(\omega + \nu) + \cos \Omega \cos(\omega + \nu) \cos i \right]$$

$$\frac{\partial z}{\partial M} = \frac{zs}{r} + r \frac{\partial \nu}{\partial M} \cos(\omega + \nu) \sin i$$

$$\text{where } s = \frac{ae \sin \nu}{\sqrt{1 - e^2}}$$

Reference: Battin, R. H.: Astronautical Guidance, McGraw-Hill Book Company, Inc., New York, 1964.

TARFRL Flow Chart



SUBROUTINE THPOSM

PURPOSE: TO FIND MINIMUM OF FUNCTION ON A GIVEN INTERVAL
BY CUBIC INTERPOLATION

ARGUMENT: ALPHA I FRACTION OF INTERVAL FROM 0. TO AMBDA
AT WHICH THIRD FUNCTION VALUE IS FOUND

AMBDA I UPPER END OF SEARCH INTERVAL AT WHICH
FUNCTION VALUE IS FOUND (LOWER END IS 0.)

FALAM I VALUE OF FUNCTION AT ALPHA TIMES AMBDA

FLAM I VALUE OF FUNCTION AT AMBDA

FOP I DERIVATIVE FUNCTION VALUE AT ABSCISSA OF 0.

F0 I FUNCTION VALUE AT ABSCISSA VALUE OF 0.

XMIN 0 ABSCISSA VALUE AT MINIMUM OF FITTED
CUBIC POLYNOMIAL

SUBROUTINES SUPPORTED: GAUSLS

A1 COEFFICIENT OF QUADRATIC TERM IN
FITTED CUBIC POLYNOMIAL

DISC DISCRIMINANT OF QUADRATIC EQUATION FORMED
BY SETTING DERIVATIVE OF CUBIC POLYNOMIAL
TO ZERO

THRA0 THREE TIMES COEFFICIENT OF CUBIC TERM
IN FITTED CUBIC POLYNOMIAL

THPØSM Analysis

THPØSM locates by cubic interpolation the minimum of a scalar function f of one variable on the closed interval from 0 to λ (positive), assuming f has a unique point on that interval where its derivative vanishes and that this extreme value is indeed a minimum. The algorithm fits a cubic polynomial through function values at 0, $\alpha\lambda$, and λ as well as through the function's slope at 0. Here $\alpha\lambda$ should be on the open interval from 0 to λ and preferably near the middle. The abscissa of the minimum is then approximately by the abscissa of the corresponding minimum of the cubic.

The coefficients of the approximating cubic can readily be derived once and for all and cast into a form facilitating speedy execution. This approach proves much more economical of machine time than solving for them each pass with a linear system routine as is frequently done in polynomial fitting.

Denote the approximating cubic polynomial by

$$c(X) = a_0 X^3 + a_1 X^2 + a_2 X + a_3. \quad (1)$$

Then clearly

$$a_3 = f(0) \quad (2)$$

and

$$a_2 = f'(0) \quad (3)$$

One also has that

$$f(\lambda) = a_0 \lambda^3 + a_1 \lambda^2 + f'(0)\lambda + f(0) \quad (4)$$

and

$$f(\alpha\lambda) = a_0 \alpha^3 \lambda^3 + a_1 \alpha^2 \lambda^2 + f'(0)\alpha\lambda + f(0). \quad (5)$$

Solving these last two equations for a_0 and a_1 yields

$$a_0 = \frac{1}{\lambda^3 \alpha^2} \left[\lambda f'(0)\alpha + f(0)(1+\alpha) + \frac{\alpha^2 f(\lambda) - f(\alpha\lambda)}{1-\alpha} \right] \quad (6)$$

and

$$a_1 = \frac{1}{\lambda^2 \alpha^2} \left[\frac{f(\alpha\lambda) - \alpha^3 f(\lambda)}{1 - \alpha} - \lambda \alpha (1 + \alpha) f'(0) - f(0) (1 + \alpha + \alpha^2) \right]. \quad (7)$$

For an extreme value, X_e , one knows that $c'(X_e) = 0$, that is,

$$3a_0 X_e^2 + 2a_1 X_e + a_2 = 0. \quad (8)$$

Hence

$$X_{e1,2} = \frac{-a_1 \pm \sqrt{a_1^2 - 3a_0 a_2}}{3a_0}.$$

The question remains as to which of these two extrema is a minimum. It is shown in elementary calculus that an extremum is a minimum if, and only if, the second derivative is positive there. Now

$$c''(X) = 6a_0 X + 2a_1.$$

Hence

$$\begin{aligned} c''(X_{e1,2}) &= 6a_0 \left[\frac{-a_1 \pm \sqrt{a_1^2 - 3a_0 a_2}}{3a_0} \right] + 2a_1 \\ &= \pm 2 \sqrt{a_1^2 - 3a_0 a_2}. \end{aligned}$$

Thus the cubic has its minimum at

$$X_{\min} = \frac{-a_1 + \sqrt{a_1^2 - 3a_0 a_2}}{3a_0}.$$

The preceding formula for the minimum will obviously be inadequate when $a_0 = 0$ as is the case when minimizing a quadratic.

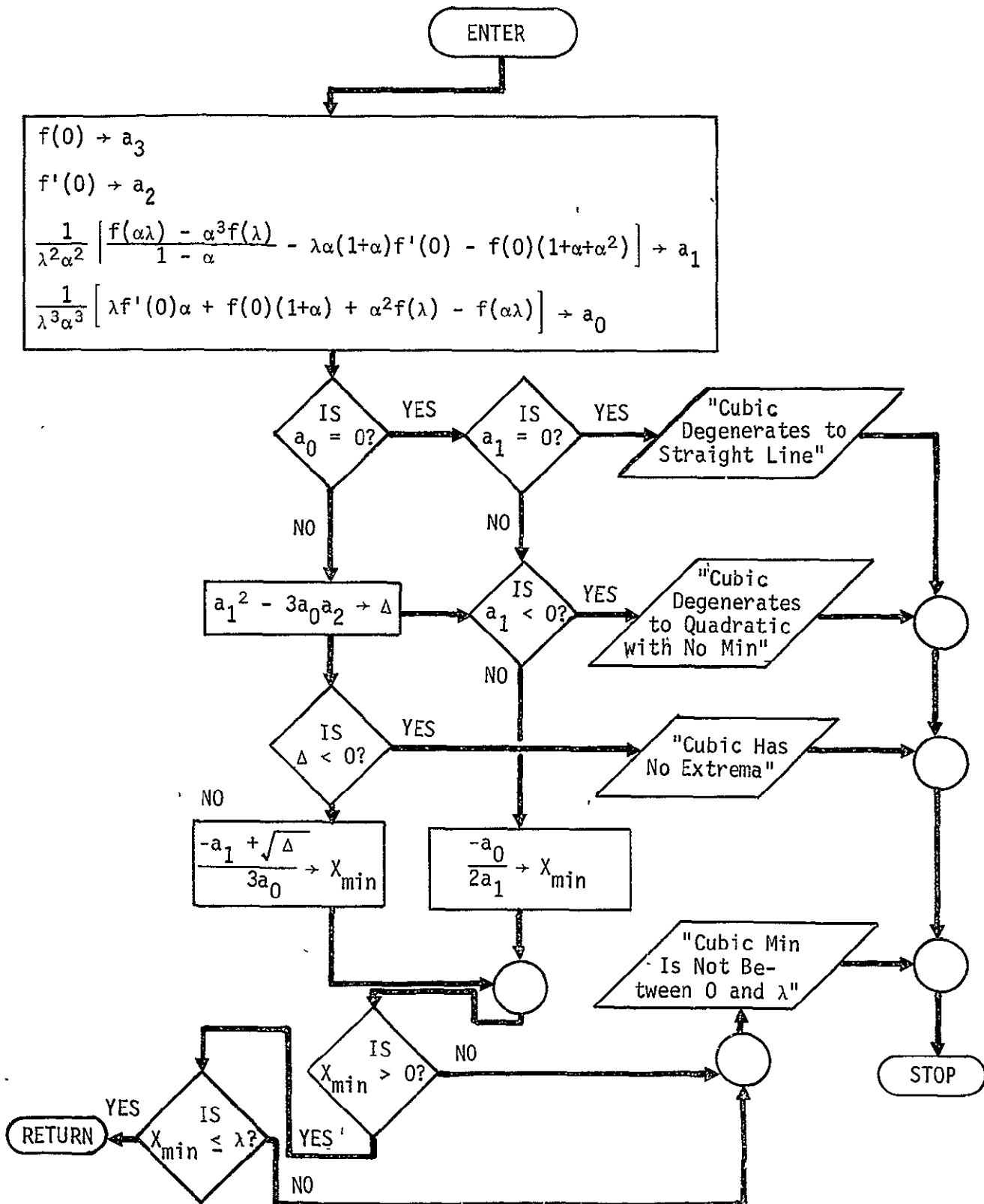
However it can be shown that

$$\lim_{a_0 \rightarrow 0} X_{\min} = -\frac{a_2}{2a_1}$$

and hence when $a_0 = 0$ we take

$$X_{\min} = -\frac{a_2}{2a_1}$$

THPØSM provides diagnostic printouts and stops execution of the calling program in the three conceivable cases of difficulty: (1) the cubic degenerates to a straight line, (2) the cubic degenerates to a quadratic with no minimum, (3) the cubic is nondegenerate but has no extrema, or (4) the minimum of the cubic does not fall between 0 and λ .



SUBROUTINE TIME

PURPOSE: TO COMPUTE THE JULIAN DATE, EPOCH 1900, FROM THE
CALENDAR DATE OR TO COMPUTE THE CALENDAR DATE FROM THE
JULIAN DATE.

CALLING SEQUENCE: CALL TIME(DAY,IYR,MO,IDAY,IHR,MIN,SEC,ICODE)

ARGUMENT: DAY I/O JULIAN DATE, EPOCH 1900

IYR O/I CALENDAR YEAR

MO O/I CALENDAR MONTH

IDAY O/I CALENDAR DAY

IHR O/I HOUR OF THE DAY

MIN O/I MINUTE OF HOUR

SEC O/I FRACTIONAL SECONDS

ICODE I OPERATIONAL MODE
= 1, INDICATES THE JULIAN DATE IS INPUT,
CALENDAR DATE IS OUTPUT
= 0, INDICATES THE CALENDAR DATE IS INPUT,
JULIAN DATE IS OUTPUT

SUBROUTINES SUPPORTED: DATAS INPUTZ PRINT VMP GIDANS
PREPUL PRNTS4 DATA PRNTS3 PRELIM
GIDANS HELIO MULTAR

SUBROUTINES REQUIRED: NONE

LOCAL SYMBOLS: IA NUMBER OF CENTURIES
IB YEARS IN PRESENT CENTURY
IP NUMBER OF MONTH (BASED ON MARCH AS NUMBER
ZERO)
IQ NUMBER OF YEARS
IR NUMBER OF CENTURIES DIVIDED BY 4
IS NUMBER OF YEARS SINCE LAST 400 YEAR
SECTION BEGAN
IT NUMBER OF LEAP YEARS IN PRESENT CENTURY
IU NUMBER OF YEARS SINCE LAST LEAP YEAR
IV NUMBER OF DAYS IN LAST YEAR

PRECEDING PAGE BLANK NOT FILMED

IX	INTERMEDIATE INTEGER
J	INTERMEDIATE INTEGER
JD	NUMBER OF DAYS IN JULIAN DATE
P	JULIAN DATE
R	FRACTIONAL PORTION OF DAY IN JULIAN DATE

SUBROUTINE TITLE

PURPOSE: TO PRINT TITLES FOR ERRAN.

CALLING SEQUENCE: CALL TITLE(LINES,TEVN,ICODE)

ARGUMENT: LINES NOT USED

TEVN I EVENT TIME

ICODE I EVENT CODE

SUBROUTINES SUPPORTED: SETEVN

LOCAL SYMBOLS: TPT TIME PREDICTING TO

COMMON USED: IPR08 NPE TPT2

SUBROUTINE TITLES

PURPOSE: TO PRINT TITLES FOR SIMUL.

CALLING SEQUENCE: CALL TITLES(TEVN,ICODE)

ARGUMENT: TEVN I EVENT TIME

ICODE I EVENT CODE

SUBROUTINES SUPPORTED: SETEVS

LOCAL SYMBOLS: TPT TIME PREDICTING TO

COMMON USED: IPROB NPE TPT2

SUBROUTINE TPPROP

PURPOSE: PROPAGATE THREE MINIPROBE TRAJECTORIES ACCORDING
 TO EITHER CONIC OR VIRTUAL - MASS MODEL

ARGUMENT: PSI I CONSTRAINT VECTOR
 UCNTL 0 CONTROL VECTOR

SUBROUTINES SUPPORTED: TPRTRG

SUBROUTINES REQUIRED: CAREL CONCAR DIMPCP MATPY SAOCS
 SCAR SPHIMF STIMP VMP

LOCAL SYMBOLS: ANPTP SEMI-MAJOR AXIS OF I-TH PROBE CONIC
 ATP INTERMEDIATE VARIABLE USED TO FIND UCNTL
 AUTKM CONVERSION FACTOR FROM AU TO KM
 BM MAGNITUDE OF B-VECTOR OF I-TH CONIC
 CSRRA COSINE OF ROLL RELEASE ANGLE OF I-TH PROBE
 CSTITP COSINE AT IMPACT OF TRUE ANOMALY
 CSTP03 INTERMEDIATE VARIABLE
 DBR INTERMEDIATE VARIABLE
 DBT INTERMEDIATE VARIABLE
 DJEDUM JULIAN DATE EPOCH 1900
 DTPITP TIME FROM PERIAPSIS TO IMPACT OF I-TH CONIC
 DTPRTP TIME FROM PERIAPSIS TO EXTRAPOLATION STATE
 DTR CONVERSION FACTOR FROM DEGREES TO RADIANS
 DTS CONVERSION FACTOR FROM DAYS TO SECONDS
 EMS1 MISS DISTANCE SQUARED FROM SECOND
 PROBE TO FIRST TARGET SITE TO WHICH
 PROBE ONE IS NOT TARGETED
 EMS2 MISS DISTANCE SQUARED FROM SECOND PROBE
 TO SECOND TARGET SITE TO WHICH PROBE
 ONE IS NOT TARGETED
 ENPTP ECCENTRICITY OF I-TH PROBE CONIC
 FACTR INTERMEDIATE VARIABLE

ICL2S SAVED VALUE OF VMP FLAG ICL2
 INMIN1 INDEX OF FIRST SITE TO WHICH PROBE
 ONE IS NOT TARGETED
 INMIN2 INDEX OF SECOND SITE TO WHICH PROBE
 ONE IS NOT TARGETED
 ISITE INDEX OF DESIRED I-TH PROBE TARGET SITE
 ISP2 CALL ARGUMENT TO VMP
 I INDEX
 NTPI INDEX FOR F AND V ARRAY USAGE
 ONE CONSTANT = 1.
 PNPTP SEMI-LATUS RECTUM OF I-TH PROBE CONIC
 PPNPTP UNIT ECLIPTIC PERIAPSIS VECTOR OF CONIC
 QQNPTP UNIT VECTOR NORMAL TO PPNPTP IN ORBIT PLANE
 RTD CONVERTS RADIANs TO DEGREEs
 RTPIP UNIT ECLIPTIC VECTOR TO I-TH PROBE SITE
 RTPISS PROBE-SPHERE UNIT VECTOR TO I-TH PROBE SITE
 RV INTERMEDIATE VECTOR
 SNRRA SINE OF ROLL RELEASE ANGLE OF I-TH PROBE
 SNTITP SINE OF TRUE ANOMALY OF CONIC AT IMPACT
 SNTPO3 INTERMEDIATE VARIABLE
 SPHERS S.O.I. RADIUS OF TARGET PLANET
 SV UNIT VECTOR NORMAL TO RV
 TM2 CONSTANT = 1.E-2
 TP2 CONSTANT = 1.E+2
 TRIMS TIME OF PROBE RELEASE IN DAYS RELATIVE
 TO START OF VMP TRAJECTORY
 TV CROSS PRODUCT OF SV AND RV
 TWO CONSTANT = 2.

VTANG TANG. VELOCITY OF ALL PROBES AT RELEASE

VTPIP ECLIPTIC VELOCITY VECTOR OF I-TH PROBE

VTPRPA ECLIPTIC VELOCITY VECTOR OF I-TH PROBE
AT RELEASE OR AT BEGINNING OF THE
CONIC EXTRAPOLATION

WWNPTP CROSS PRODUCT OF PPNPTP AND QQNPTP

XSF FINAL HELIOCENTRIC ECLIPTIC STATE OF
I-TH PROBE RETURNED BY VMP IN KM,KM/SEC

XSI INITIAL HELIOCENTRIC ECLIPTIC STATE OF
I-TH PROBE RELEASE PROPAGATED BY VMP

ZERO CONSTANT = 0.

COMMON COMPUTED/USED:	ICL2 VSCRPA	RSCRPA	SPHERE	TRTM	VIMTP
COMMON COMPUTED:	AATTP FPATP TM	ADCTP ICL	ALNGTH INCMT	ARATP IPRINT	DJEITP ISPH
COMMON USED:	ACTPP CSRASA F ISAO RSCRHA TRANSF	BDR DCTP GMUP NBOD RSI VSCRHA	BDT DELTM IFIN2 NB RTPS VSI	B DJERN IMIN NTP SNDCSA V	CSOCSA DSI IFROP RATP SNRASA WFLS

TPPRØP Analysis

The subroutine TPPRØP has the sole responsibility for propagating miniprobes. It is called on to do so in two basic applications. The first is in calculating the miniprobe targeting constraint ψ given the release control \underline{u} as repeatedly required in the miss-minimization process (see analysis of TPRTRG). The second is in describing the minimum-miss miniprobe approach trajectories once the optimal release controls have been found. This includes generating impact data for both conic and virtual-mass propagation as well as time histories of the latter. The logic of TPPRØP is considerably complicated by the requirement that it be able to propagate the approach trajectories according to either a high-speed conic model or an accurate virtual-mass n-body integrator.

First consider the problem of calculating ψ given \underline{u} . For this application of TPPRØP, the miss-minimization status flag, IFIN2, is set to 1. In either propagation mode the velocity, \underline{v}_{iR} , of the i th miniprobe just after release must be computed first of all. Let \underline{v}_{BR} be the velocity of the bus at release and ϕ_i be the release roll angle of the i th probe. Let \underline{U} , \underline{V} and \underline{H} be the probe release reference vectors defined in the analysis section of TPRTRG. By ϕ_i we mean the angle the release velocity increment of the i th miniprobe makes with the \underline{U} direction. It should not be confused with the angle the i th probe arm makes with the \underline{U} direction, which is $\pi/2$ radians smaller. Next define v_T to be the common tangential velocity of the miniprobes at release. Then the velocity of the i th miniprobe at release is

$$\underline{v}_{iR} = v_T \left[\cos \phi_i \underline{U} + \sin \phi_i \underline{V} \right] + \underline{v}_{BR} \quad i = 1, 2, 3. \quad (1)$$

Let α_H and δ_H denote the right ascension and declination, respectively, of the spin axis at release. Then by expressing \underline{U} and \underline{V} in terms of these angles, equation (1) reduces to the following in the planetocentric ecliptic system:

$$\underline{v}_{iR} = v_T \left[\cos \phi_i \begin{pmatrix} \sin \alpha_H \\ -\cos \alpha_H \\ 0 \end{pmatrix} + \sin \phi_i \begin{pmatrix} \cos \alpha_H \sin \delta_H \\ \sin \alpha_H \sin \delta_H \\ -\cos \delta_H \end{pmatrix} \right] + \underline{v}_{BR} \quad i = 1, 2, 3 \quad (2)$$

Simplifying equation (2) yields the computational form of the ith miniprobe velocity increment:

$$\underline{v}_{iR} = v_T \begin{pmatrix} \sin \alpha_H \cos \phi_i + \cos \alpha_H \sin \delta_H \sin \phi_i \\ -\cos \alpha_H \cos \phi_i + \sin \alpha_H \sin \delta_H \sin \phi_i \\ -\cos \delta_H \sin \phi_i \end{pmatrix} + \underline{v}_{BR} \quad (3)$$

$$i = 1, 2, 3.$$

The sines and cosines of α_H and δ_H necessary in equation (3) are all calculated in a single call to the subroutine SAØCS given the spin-axis orientation flag, ISAØ.

At this point the algorithms for calculating the constraint ψ diverge for the two miniprobe propagation models. In the conic propagation mode, signaled by the flag IPRØP set to 1, \underline{v}_{BR} in equation (3) represents the velocity of the equivalent conic release state of the bus (see TPRTRG analysis). Hence, to determine the actual B-plane pierce point coordinates $(\underline{B}_i \cdot \underline{T}_i)_A$ and $(\underline{B}_i \cdot \underline{R}_i)_A$ as well as the parameters \underline{S} , \underline{T} , \underline{R} and a for the i th miniprobe, TPPRØP simply applies the subroutine STIMP to the state $\begin{pmatrix} \underline{r}_{BR}^T & \underline{T}^T \\ \underline{v}_{iR}^T & \underline{T}^T \end{pmatrix}$ where \underline{r}_{BR} denotes the equivalent conic position of the bus (and hence also of the i th probe) at release. Determining the desired B-plane pierce point coordinates, $(\underline{B}_i \cdot \underline{T}_i)_D$ and $(\underline{B}_i \cdot \underline{R}_i)_D$, of the i th probe is complicated by the fact that the miniprobes must be correctly paired with the impact sites. The first miniprobe is targeted to the miniprobe site whose pierce point at the time of the calculation of the initial control estimate in TPRTRG was nearest the bus pierce point. Hence $(\underline{B}_1 \cdot \underline{T}_1)_D$ and $(\underline{B}_1 \cdot \underline{R}_1)_D$ are readily calculated by the appropriate call to DIMPCP, with the right ascension and declination of the target site used in the initial control estimate and the \underline{S} and a of the current miniprobe 1 trajectory. Next TPRTRG computes two sets of desired B-plane coordinate pairs, $(\underline{B}_2 \cdot \underline{T}_2)_D$ and $(\underline{B}_2 \cdot \underline{R}_2)_D$, by applying DIMPCP successively to each of the remaining pairs of miniprobe target sites using in both cases the

\underline{S} and \underline{a} of the current miniprobe 2 trajectory. With these two sets of desired pierce point coordinate pairs relative to miniprobe 2 now available for comparison, TPPRØP selects the set whose Euclidean distance from the pierce point of miniprobe 2 is the smaller. Finally, $(\underline{B}_3 \cdot \underline{T}_3)_D$ and $(\underline{B}_3 \cdot \underline{R}_3)_D$ are calcu-

lated by calling DIMPCP with right ascension and declination of the remaining miniprobe target site and the \underline{S} and \underline{a} of the current miniprobe 3 trajectory. At this point an approximately inherent in the application of the subroutine DIMPCP to the miniprobes must be noted. The time-varying transformation from planetocentric-ecliptic coordinates to the probe-impact frame required by DIMPCP is held fixed at the time of the bus impact. This approximation is more than adequate for engineering purposes. All of the required B-plane data for the three miniprobes having been assembled, TPPRØP can now calculate the i th and $(i+3)$ rd components of the constraint vector:

$$\begin{aligned}\psi_i &= C_i \left[(\underline{B}_i \cdot \underline{T}_i)_D - (\underline{B}_i \cdot \underline{T}_i)_A \right] \\ \psi_{i+3} &= C_i \left[(\underline{B}_i \cdot \underline{R}_i)_D - (\underline{B}_i \cdot \underline{T}_i)_A \right] \quad i = 1, 2, 3.\end{aligned}\quad (4)$$

Here the C_i 's are weighting factors indicating the relative importance of achieving nearby impacts at the respective miniprobe target sites. Finally, the release roll angle for the next miniprobe can be found from that of the current one by applying the addition formulas for the sine and cosine:

$$\sin \phi_{i+1} = (\sqrt{3} \sin \phi_i + \cos \phi_i) / 2 \quad (5)$$

$$\cos \phi_{i+1} = (\cos \phi_i - \sqrt{3} \sin \phi_i) / 2. \quad (6)$$

The iterative process is started by noting that ϕ_1 is simply ϕ , the first component of the control vector.

For virtual-mass propagation indicated by IPRØP=2, the general structure of the $\underline{\psi}$ computation remains the same but the interpretation of the constituent state vectors changes. The \underline{v}_{BR} in equation (3) is taken as the velocity of the actual virtual-mass release state. Then VMP is called to integrate the virtual mass state $\begin{pmatrix} \underline{r}_{BR}^T \\ \underline{v}_{BR}^T \end{pmatrix}$ of the i th miniprobe just after

release to a pseudosphere whose radius is one-tenth that of the actual Laplacian sphere of influence. From this distance inward, conic extrapolation of the current state suffices for engineering calculations. For the actual B-plane pierce-point coordinates, $(B_i \cdot T_i)_A$ and $(B_i \cdot R_i)_A$, as well as the quantities \underline{S} , \underline{T} , \underline{R} and

for the i th miniprobe, TPPRØP again simply makes a single call to STIMP but this time with the virtual-mass state at the pseudosphere rather than the equivalent conic state at release. The remainder of the virtual-mass constraint vector computation including the computation of the desired miniprobe B-plane pierce points via the subroutine DIMPCF proceeds exactly as for the conic model.

Next consider the provisions in TPRTRG for describing the minimum-miss miniprobe approach trajectories. Throughout this application, the miss-minimization status flag, IFIN2, must be set to 2. First in this area TPPRØP must supply the impact data for the miniprobes using the conic propagation model on the conic minimum-miss release controls. This is done by setting the propagation mode flag, IPRØP to 1. Then with IFIN2=2, TPPRØP uses the state

$\begin{pmatrix} \underline{r}_{BR}^T \\ \underline{v}_{iR}^T \end{pmatrix}^T$ constructed from the bus equivalent conic release state, the conic minimum-miss release controls, and equation (3) to generate an osculating conic by a call to CAREL. The sine and cosine of the true anomaly and the time at impact are determined by a call to SPHIMP. Then the Cartesian planetocentric ecliptic state is evaluated by a call to CONCAR. From these data the right ascension and declination of the impact site, as well as the time, speed, and flightpath angle at impact for the i th miniprobe can be calculated from the formulas used in computing the same quantities for the bus. These are described in the TPRTRG analysis. The angle of attack, α_i , for the i th miniprobe however requires separate treatment. It is assumed that the longitudinal body axes of each miniprobe remains parallel until impact to the spacecraft spin axis at release. Thus if \underline{v}_{iI} represents the velocity of the i th miniprobe at impact,

$$\alpha_i = \cos^{-1} \left(\underline{h} \cdot \underline{v}_{iI} / \|\underline{v}_{iI}\| \right) \quad i = 1, 2, 3. \quad (7)$$

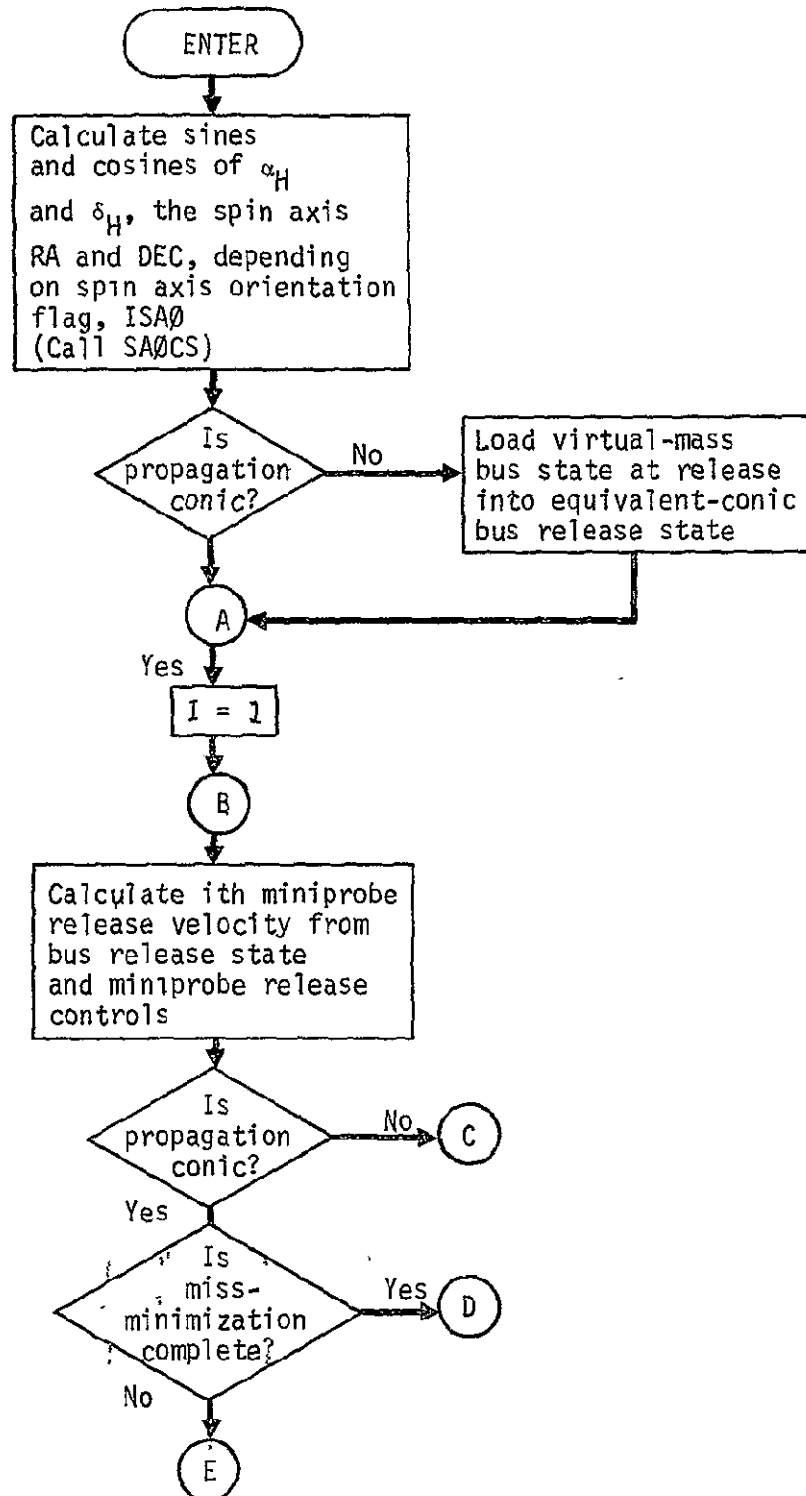
Equation (5) can be expressed in terms of the angles α_H and δ_H as follows:

$$\alpha_i = \cos^{-1} \left\{ \left(\cos \delta_H \left[(v_{iI})_1 \cos \alpha_H + (v_{iI})_2 \sin \alpha_H \right] + (v_{iI})_3 \sin \delta_H \right) / \|v_{iI}\| \right\} \quad i = 1, 2, 3. \quad (8)$$

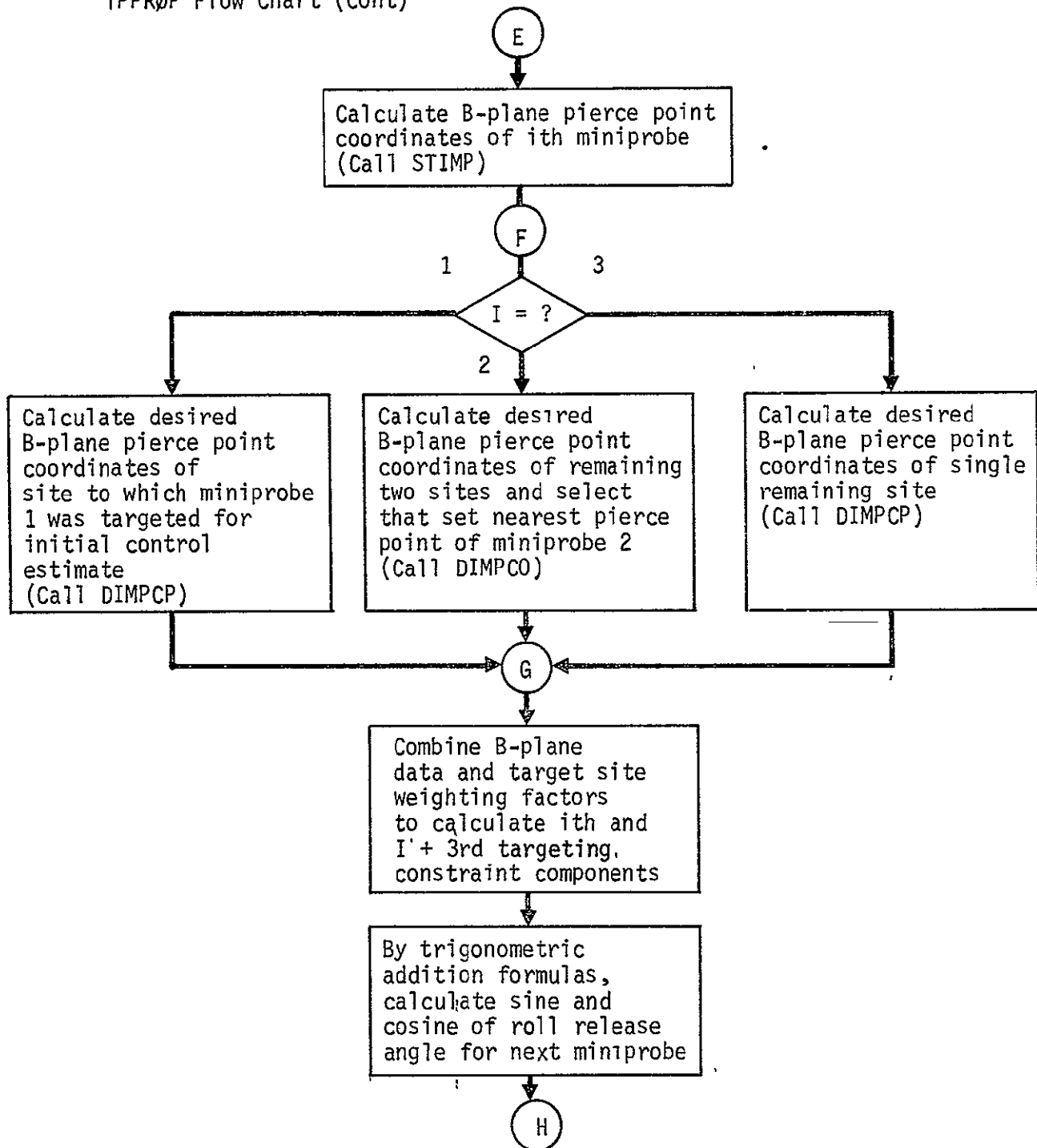
The second trajectory-descriptive function of TPPRØP is calculating the virtual-mass miniprobe approach-trajectory time histories and impact data. This is always done for the conic minimum-miss release controls and also for the virtual-mass controls whenever these are calculated. The propagation flag, IPRØP is simply set to 2 and the heading "Miniprobe I Minimum-Miss Approach Trajectory" is printed. Then with IFIN2=2, the state

$\begin{pmatrix} \underline{r}_{BR}^T & | & \underline{v}_{iR}^T \end{pmatrix}^T$ produced from the bus actual virtual-mass release state, the appropriate minimum-miss release controls (conic or virtual-mass as the case may be), and equation (3) is integrated all the way to impact by VMP with its print flag on and its print increments set at 5 days and 100 integration steps. In this simple manner the miniprobe approach time histories are provided. Finally, the impact data for the virtual-mass ith miniprobe trajectory are calculated by the same steps as for the conic case except that the actual virtual-mass miniprobe impact state rather than the equivalent conic miniprobe release state is used in generating the osculating conic via the call to CAREL.

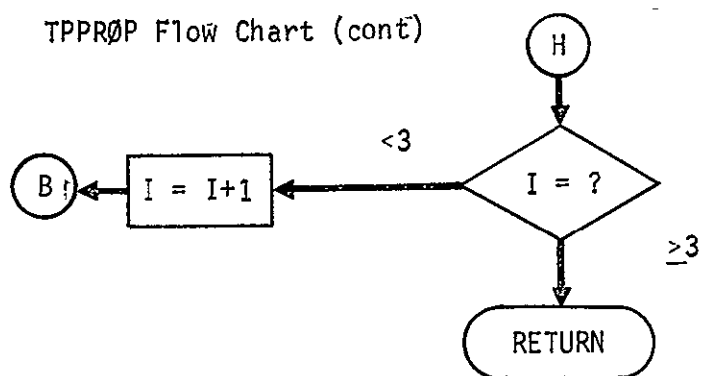
TPPROP Flow Chart



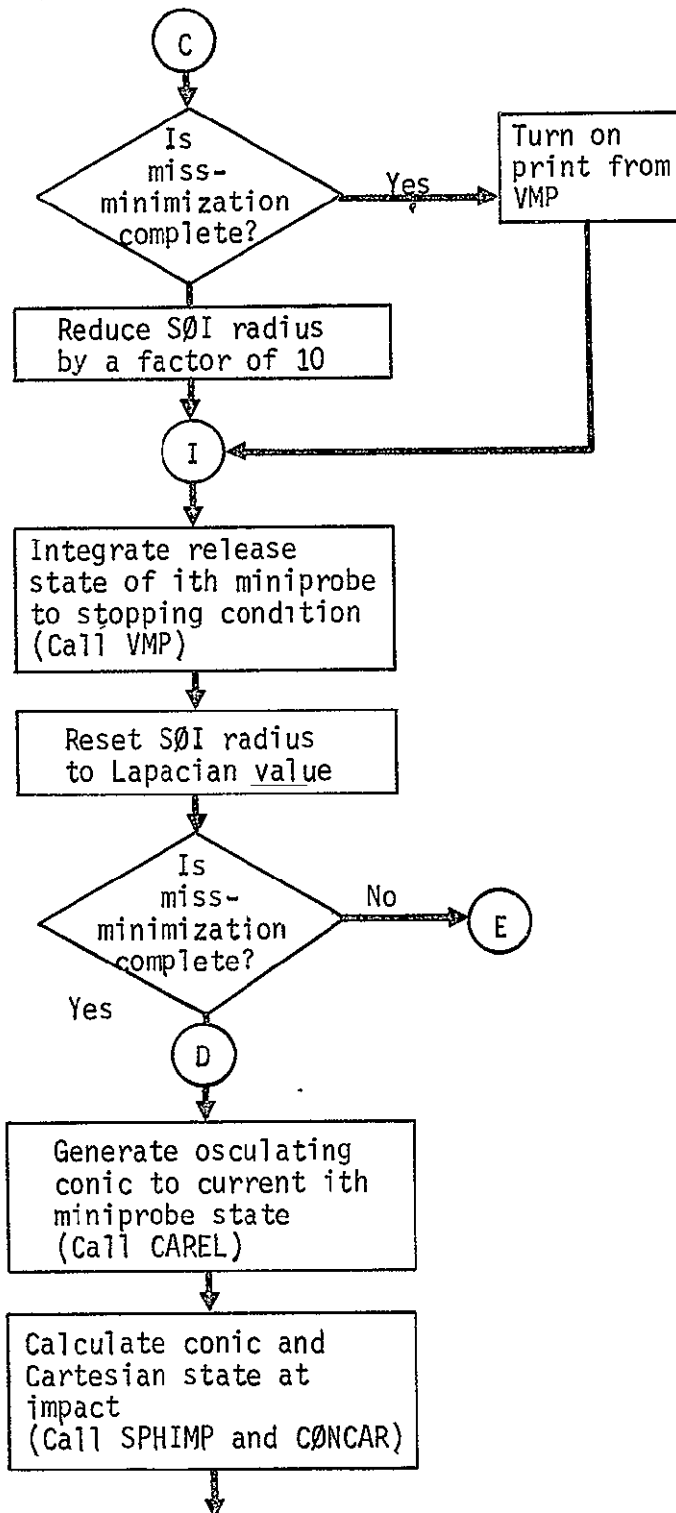
TPPRØP Flow Chart (cont)



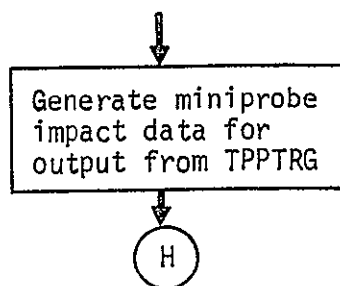
TPPRØP Flow Chart (cont)



TPPRØP Flow Chart (cont)



TPPRØP Flow Chart (concl)



SUBROUTINE TPRTRG

PURPOSE: TO CONTROL MINIPROBE TARGETING PROCEDURE

SUBROUTINES SUPPORTED: GIDANS

SUBROUTINES REQUIRED:	CAREL	CONCAR	DIMPCP	GAUSLS	HPOST
	MATIN	MATPY	PECEQ	SAOCS	SPHIMP
	STIMP	SUBSOL	TPPROP	VMP	

LOCAL SYMBOLS:	ANPT	SEMI-MAJOR AXIS OF BUS NEAR-PLANET OSCULATING CONIC IN KM
	AP	SEMI-MAJOR AXIS OF PERTURBED BUS NEAR- PLANET OSCULATING CONIC IN KM
	ASC	SAME AS ANPT
	BDRP	PROJECTION ON RVP OF PLANETOCENTRIC ECLIPTIC VECTOR TO IMPACT PLANE ASYMPTOTE PIERCE POINT OF PERTURBED NEAR-PLANET OSCULATING CONIC IN KM
	BDRSC	PROJECTION OF RVSC OF PLANETOCENTRIC ECLIPTIC VECTOR TO IMPACT PLANE ASYMPTOTE PIERCE POINT OF BUS NEAR-PLANET OSCULATING CONIC IN KM
	BDTP	PROJECTION ON TVP OF PLANETOCENTRIC ECLIPTIC VECTOR TO IMPACT PLANE ASYMPTOTE PIERCE POINT OF PERTURBED BUS NEAR-PLANET OSCULATING CONIC IN KM
	BDTSC	PROJECTION ON TVSC OF PLANETOCENTRIC ECLIPTIC VECTOR TO IMPACT PLANE ASYMPTOTE PIERCE POINT OF BUS NEAR-PLANET OSCULATING CONIC IN KM
	BP	MAGNITUDE OF PLANETOCENTRIC VECTOR TO IMPACT PLANE ASYMPTOTE FIERCE POINT OF PERTURBED BUS NEAR-PLANET OSCULATING CONIC IN KM
	BSC	MAGNITUDE OF PLANETOCENTRIC VECTOR TO IMPACT PLANE ASYMPTOTE PIERCE POINT OF BUS NEAR-PLANET OSCULATING CONIC IN KM
	CSTISC	COSINE OF TRUE ANAMOLY ON BUS NEAR-PLANET OSCULATING CONIC AT IMPACT
	CSTRSC	COSINE OF TRUE ANAMOLY ON BUS NEAR-PLANET OSCULATING CONIC AT EQUIVALENT RELEASE

STATE

DBRTP ARRAY OF DESIRED RVSC-AXIS IMPACT PLANE
COORDINATES FOR THREE MINIPROBES TO
ACHIEVE RESPECTIVE THREE MINIPROBE TARGET
SITES

DBTTP ARRAY OF DESIRED TVSC-AXIS IMPACT PLANE
COORDINATES FOR THREE MINIPROBES TO
ACHIEVE RESPECTIVE THREE MINIPROBE TARGET
SITES

DCSAF ECLIPTIC DECLINATION OF SPIN AXIS IN DEG
FOR FIXED ORIENTATION SPIN AXIS MODE

DCSC DECLINATION IN DEG OF BUS IMPACT SITE
RELATIVE TO PLANETOCENTRIC PROBE-SPHERE
FRAME

DDCS MINIMUM-MISS ECLIPTIC DECLINATION OF SPIN
AXIS IN DEG

DELTPS SAVED VALUE OF TIME DURATION BETWEEN VMP
PRINTOUTS

DELTUV COMMON VELOCITY INCREMENT LENGTH IN KM/SEC
USED IN U AND V DIRECTIONS IN
APPROXIMATING JACOBIAN MATRIX SENSM BY
DIVIDED DIFFERENCES

DJECA JULIAN DATE EPOCH 1900 ON BUS NEAR-PLANET
OSCULATING CONIC AT CLOSEST APPROACH

DJEISC JULIAN DATE EPOCH 1900 ON BUS NEAR-PLANET
OSCULATING CONIC AT IMPACT

DJENPS JULIAN DATE EPOCH 1900 ON BUS NEAR-PLANET
OSCULATING CONIC AT EXTRAPOLATION STATE

DJEPOC JULIAN DATE ON JANUARY .5 1900 E.T.

DRAS MINIMUM-MISS ECLIPTIC RIGHT ASCENSION
OF SPIN AXIS IN DEG

DRRA MINIMUM-MISS RELEASE ROLL ANGLE OF
MINIPROBE 1 IN DEG

DTPISC TIME INTERVAL IN SEC FROM PERIAPSIS TO
IMPACT ON BUS NEAR-PLANET OSCULATING CONIC

DTPNPS TIME INTERVAL IN SEC FROM PERIPSIS TO
EXTRAPOLATION STATE ON BUS NEAR-PLANET
OSCULATING CONIC

CONVERSION FACTOR FROM DEGREES TO RADIANS

DTS CONVERSION FACTOR FROM DAYS TO SECONDS

EMSMN SQUARE OF DISTANCE BETWEEN ASYMPTOTE
PIERCE POINTS IN IMPACT PLANE OF BUS AND
NEAREST MINIPROBE IN KM^2

EMS SQUARE OF DISTANCE BETWEEN ASYMPTOTE
PIERCE POINTS IN IMPACT PLANE OF BUS AND
ITH MINIPROBE IN KM^2

ENPT ECCENTRICITY OF BUS NEAR-PLANET OSCULATING
CONIC

EPSLS UPPER BOUND ON WEIGHTED SUM OF CHANGE IN
LENGTH OF CONTROL VECTOR AND CHANGE IN
MAGNITUDE OF MISS INDEX USED IN
CONVERGENCE CRITERION FOR LEAST-SQUARES
ROUTINE

FPASC FLIGHT PATH ANGLE OF BUS IN DEG AT IMPACT
ON NEAR-PLANET OSCULATING CONIC

ICLS SAVED VALUE OF VMP FLAG ICL INDICATING
WHETHER OR NOT TRAJECTORY HAS REACHED
CLOSEST APPROACH

ICL2S SAVED VALUE OF VMP FLAG ICL2 INDICATING
WHETHER OR NOT TRAJECTORY IS TO BE STOPPED
AT CLOSEST APPROACH

ICONV1 CONVERGENCE INDICATOR FOR LEAST-SQUARES
ROUTINE
=1 CONVERGENCE
=2 NO CONVERGENCE

IEPHEMS SAVED VALUE OF VMP FLAG IEPHEM INDICATING
WHETHER OR NOT ORB IS TO BE CALLED BEFORE
CALLING EPHEM

INPRS SAVED VALUE OF VMP VARIABLE INPR
INDICATING NUMBER OF INTEGRATION STEPS
BETWEEN PRINTOUTS

IPCSK FLAG INDICATING THE PLANETOCENTRIC PROBE-
SPHERE COORDINATE SYSTEM OPTION
=1 EQUATORIAL
=2 SUBSOLAR ORBIT PLANE

IPRNTS SAVED VALUE OF VMP FLAG IPRINT INDICATING
WHETHER OR NOT PRINTOUT IS TO OCCUR

I_{PROPI} FLAG INDICATING WHETHER CONIC OR VIRTUAL-
MASS MINIPROBE PROPAGATION IS DESIRED
=1 CONIC
=2 VIRTUAL-MASS

I_{SAOP} MODIFIED SPIN AXIS ORIENTATION MODE FLAG
USED IN ORIENTING SPIN AXIS FOR GENERATING
INITIAL RELEASE CONTROL ESTIMATE

I_{SPHS} SAVED VALUE OF VMP FLAG I_{SPH} INDICATING
WHETHER OR NOT SPHERE OF INFLUENCE HAS
BEEN PIERCED

N_{CNTRL} NUMBER OF RELEASE CONTROLS

P_{NP} SEMI-LATUS RECTUM OF BUS NEAR-PLANET
OSCULATING CONIC

P_{PNPT} PLANETOCENTRIC ECLIPTIC UNIT VECTOR IN
DIRECTION OF PERIAPSIS OF BUS OSCULATING
NEAR-PLANET CONIC

P_{SI} IMPACT PLANE CONSTRAINT VECTOR

Q_{QNPT} PLANETOCENTRIC ECLIPTIC UNIT VECTOR LYING
IN PLANE OF NEAR-PLANET OSCULATING CONIC
90 DEG ADVANCED FROM P_{PNPT}

R_{ASAF} ECLIPTIC RIGHT ASCENSION OF SPIN AXIS FOR
FIXED ORIENTATION SPIN AXIS MODE

R_{ASC} RIGHT ASCENSION OF BUS IMPACT SITE IN DEG
RELATIVE TO PLANETOCENTRIC PROBE-SPHERE
FRAME

R_{CM} MAGNITUDE OF PLANETOCENTRIC POSITION
VECTOR OF BUS AT IMPACT IN KM

R_{PVEC} PLANETOCENTRIC ECLIPTIC UNIT VECTOR TO BUS
IMPACT SITE

R_{PVSS} PLANETOCENTRIC SUBSOLAR UNIT VECTOR TO BUS
IMPACT POINT

R_{SCNPS} PLANETOCENTRIC ECLIPTIC POSITION VECTOR
OF BUS EXTRAPOLATION STATE IN KM

R_{TD} CONVERSION FACTOR FROM RADIANS TO DEGREES

R_{VP} PLANETOCENTRIC ECLIPTIC UNIT VECTOR IN
DIRECTION OF CROSS PRODUCT OF ASYMPTOTE
OF NEAR-PLANET OSCULATING CONIC FOR

PERTURBED BUS BY ECLIPTIC POLE VECTOR

RVSC PLANETOCENTRIC ECLIPTIC UNIT VECTOR IN
DIRECTION OF CROSS PRODUCT OF ASYMPTOTE
OF NEAR-PANET OSCULATING CONIC FOR BUS
BY ECLIPTIC POLE VECTOR

SENSM JACOBIAN SENSITIVITY MATRIX OF BUS NEAR-
PLANET OSCULATING CONIC ASYMPTOTE
PIERCE POINT COORDINATES TO VELOCITY
INCREMENTS AT RELEASE IN THE U AND V
DIRECTIONS

SNTISC SINE OF TRUE ANAMOLY ON BUS NEAR-PLANET
OSCULATING CONIC AT IMPACT

SNTRSC SINE OF TRUE ANAMOLY ON BUS NEAR-PLANET
OSCULATING CONIC AT EQUIVALENT RELEASE
STATE

SPHERS SAVED VALUE OF RADIUS OF SOI FOR TARGET
PLANET IN ASTRONOMICAL UNITS

SVP PLANETOCENTRIC ECLIPTIC UNIT VECTOR IN
DIRECTION OF ASYMPTOTE OF NEAR-PLANET
OSCULATING CONIC OF PERTURBED BUS

SVSC PLANETOCENTRIC ECLIPTIC UNIT VECTOR IN
DIRECTION OF ASYMPTOTE OF NEAR-PLANET
OSCULATING CONIC OF BUS

S0 MAXIMUM PERMISSIBLE LENGTH OF PSEUDO-
INVERSE CONTROL STEP

TARGM NEWTON-RAPHSON MATRIX FOR TARGETING BUS
TO NEAREST MINIPROBE TARGET SITE

TM3 CONSTANT 1.0E-03

TM4 CONSTANT 1.0E-04

TM5 CONSTANT 1.0E-05

TP20 CONSTANT 1.0E+20

TP2 CONSTANT 1.0E+02

TP4 CONSTANT 1.0E+04

TVP CROSS PRODUCT OF SVP BY RVP

TVSC CROSS PRODUCT OF SVSC BY RVSC

UCNTRL RELEASE CONTROL VECTOR

UTPR APPROXIMATE VELOCITY INCREMENT IN U
DIRECTION AT RELEASE IN KM/SEC NECESSARY
TO TARGET BUS TO NEAREST MINIPROBE TARGET
SITE

VCM MAGNITUDE OF PLANETOCENTRIC VELOCITY OF
BUS AT IMPACT IN KM/SEC

VSCNPS PLANETOCENTRIC ECLIPTIC VELOCITY VECTOR OF
BUS EXTRAPOLATION STATE IN KM/SEC

VSCRPM MAGNITUDE OF VELOCITY OF BUS AT EQUIVALENT
CONIC RELEASE STATE IN KM/SEC

VSCRPP PLANETOCENTRIC ECLIPTIC VELOCITY VECTOR
AT EQUIVALENT CONIC RELEASE STATE IN
KM/SEC USED IN EVALUATING SENSIM

VTANG MINIMUM-MISS TANGENTIAL VELOCITY OF
MINIPROBES AT RELEASE IN KM/SEC

VTPR APPROXIMATE VELOCITY INCREMENT IN V
DIRECTION AT RELEASE IN KM/SEC NECESSARY
TO TARGET BUS TO NEAREST MINIPROBE TARGET
SITE

WWNPT PLANETOCENTRIC ECLIPTIC UNIT VECTOR IN
DIRECTION OF ANGULAR MOMENTUM FOR NEAR-
PLANET BUS OSCULATING CONIC

XSF FINAL HELIOCENTRIC ECLIPTIC BUS STATE
IN KM AND KM/SEC RETURNED BY VMP AS THE
FIRST STATE INSIDE PROBE SPHERE

XSI INITIAL HELIOCENTRIC ECLIPTIC BUS STATE IN
KM AND KM/SEC PROPAGATED TO IMPACT BY VMP

YM FINAL MINIMUM VALUE OF MISS INDEX

COMMON COMPUTED/USED:	ACTPP	DCTP	DELTM	DELTP	DJERN
	DTPRSC	GMUP	ICL2	ICL	IEPHEM
	IMIN	INPR	IPRINT	IPROP	ISAO
	ISPH	RATP	RSCRHA	RTPS	SPHERE
	VSCRHA				

COMMON COMPUTED:	IFIN2	IPCSP	KHIT	RPSP	WFLS
------------------	-------	-------	------	------	------

COMMON USED:	AATTP	ACKT	ADCTP	ALNGTH	ARATP
	CSDCSA	CSRASA	OG	DJEITP	DSI

FIVE	FPAIP	IPCS	KTAR	KUR
NINETY	NTP	ONE	PMASS	RIN
RPS	RSCRPA	RSCRPM	RSI	SNOCSA
SNRASA	TAR	TM	TOL	TRANSF
TWO	VIMTP	VSCRPA	VSI	WGHTM
ZERO				

TPRTRG Analysis

TPRTRG is the executive routine directing multiprobe targeting. To do so it must accomplish four basic tasks: (1) process miniprobe targeting input data, (2) generate initial estimates for the release controls, (3) apply the least-squares routine, GAUSLS, to minimize the miniprobe miss index, and (4) use the miniprobe propagating routine, TPPROP, to generate minimum-miss approach trajectories and impact data. Each of these objectives, as well as the routine's printout, is discussed in the following paragraphs.

The processing of the miniprobe targeting input involves three major aspects. First an equivalence must be set up between the multipurpose targeting variables of NOMINAL and the mnemonic symbolism of TPRTRG. Second, a conic miniprobe release state must be determined that is equivalent to the virtual-mass release state for use in the high-speed conic miniprobe propagation model. By equivalent we mean that although one state falls on a conic and the other on a virtual-mass trajectory, both states occur at the same absolute time and the two trajectories are tangent at impact. To generate the equivalent conic release state, the subroutine VMP is called to propagate the bus state from release to impact. Before doing this, however, the VMP trajectory condition flags are stored for subsequent restoration after exiting TPRTRG. The virtual-mass impact state is then fit with an osculating conic by a call to CAREL. The osculating conic is then propagated back to the actual release time via calls to the subroutines HPOST and CONCAR. Using this conic release state, a set of minimum-miss controls for the conic model can now readily be determined. The third phase of processing miniprobe input data, namely setting up the several targeting options, is next begun. First the coordinate transformation matrix, C, from the planetocentric inertial ecliptic coordinates to the Cartesian frame in which the miniprobe impact sites are specified is generated by a call to either the subroutine SUBSOL or PECEQ, depending on the state of the coordinate-system flag IPCSK. The spin axis orientation mode desired is stored in the flag ISAØ. It is used by TPRTRG in devising an initial control estimate and by TPPROP in all miniprobe propagations. Finally, the desired miniprobe propagation mode is stored in the flag IPRØPI. This variable is used solely by TPRTRG in deciding on which propagation mode to request from TPPROP via the common flag IPRØP.

TPRTRG next deals with the problem of generating an initial estimate of the control vector. First consider the problem of estimating the conic-model minimum-miss controls. Initial estimates for two of the controls (the spin axis orientation angles) depend of course on the spin axis orientation mode. In three of the four possible modes, the inertial-ecliptic declination and right ascension of the spin axis are fixed rather than free controls. Hence no initial estimates for them need be provided. In the remaining mode, however, both of these controls are free, and the initial estimates provided for them are simply those that bring the spin axis into coincidence with the spacecraft velocity vector at release. This orientation was chosen for the initial estimate since it produces the widest distribution of miniprobe entry sites for a given combination of the remaining two release controls. Regardless of the orientation mode, the spin axis pointing direction is specified throughout miniprobe targeting by the sines and cosines of its ecliptic declination and right ascension. These trigonometric functions are always calculated by the subroutine SAØCS from the appropriate components of the control vector and the spin axis orientation flag. Since the spin axis pointing direction is necessary in initially estimating the other two controls, TPRTRG must call SAØCS to obtain the above trigonometric characterization.

Initial estimates for the roll release angle, ϕ , of the first probe and the tangential velocity at release, v_T , can then be generated by merely targeting the first miniprobe to the miniprobe target site nearest the impact point for the bus on the trajectory existing at release. First, the B-plane pierce point of the bus is calculated by calling the subroutine STIMP with the bus impact state. Then the desired B-plane pierce point corresponding to each of the three miniprobe target sites is computed through a call to DIMPCP. It must be noted that this calculation is only approximate since it assumes all of the miniprobes have the same S and impact time as the bus. Nonetheless, the accuracy is more than sufficient for engineering purposes. Next the pierce points of the desired miniprobe impact sites are compared to find the one nearest that of the bus. The release velocity increment perpendicular to the bus spin axis that would target the first miniprobe is then approximated by a single Newton-Raphson step. Let $B_B \cdot T$, $B_B \cdot R$, $B_1 \cdot T$, and $B_1 \cdot R$ denote the B-plane pierce point components corresponding to the bus and the desired miniprobe impact sites, respectively. Define a constraint vector as

$$\psi = \begin{pmatrix} B_1 \cdot T - B_B \cdot T \\ B_1 \cdot R - B_B \cdot R \end{pmatrix} \quad (1)$$

Let \underline{H} denote a unit vector in the direction of the spin axis of the spacecraft. Using \underline{H} , define \underline{U} and \underline{V} as

$$\underline{U} = \underline{H} \times \underline{Z}_{ec} / || \underline{H} \times \underline{Z}_{ec} || \quad (2)$$

$$\underline{V} = \underline{H} \times \underline{U} \quad (3)$$

where \underline{Z}_{ec} is the inertial ecliptic pole vector. Then a convenient probe-release Cartesian frame is defined by the triple $\underline{U}-\underline{V}-\underline{H}$. Let Δv_U and Δv_V denote the components of the release velocity increment in the \underline{U} and \underline{V} directions, respectively.

Then the control vector is given by

$$\underline{X} = \begin{pmatrix} \Delta v_U \\ \Delta v_V \end{pmatrix}. \quad (4)$$

Let J denote the Jacobian matrix of $\underline{\psi}$ with respect to \underline{X} ; i.e.,

$$J_{ij} = \frac{\partial \psi_i}{\partial X_j} \quad \begin{matrix} i = 1, 2 \\ j = 1, 2. \end{matrix} \quad (5)$$

One Newton-Raphson step then approximates the targeting control vector as

$$\underline{X} = J^{-1} \underline{\psi}. \quad (6)$$

TPRTRG computes the Jacobian matrix by divided differencing. Let \underline{v}_{BR} represent the equivalent conic planetocentric velocity of the bus at release. Let δ_{U-BR}^v and δ_{V-BR}^u denote the perturbations in \underline{v}_{BR} caused by a velocity increment of magnitude δv in the \underline{U} and \underline{V} directions, respectively. Then clearly

$$\delta_{U-BR}^v = \delta v \underline{U} \quad (7)$$

$$\delta_{V-BR}^v = \delta v \underline{V}. \quad (8)$$

Let α_H and δ_H be the right ascension and declination of the spin axis, respectively. By expressing the definitions of \underline{U} and \underline{V} in terms of these angles, equations (7) and (8) can be expressed in the planetocentric ecliptic frame as

$$\delta_{\underline{U}-BR}^{\underline{V}} = \delta v (\sin \alpha, -\cos \alpha, 0)^T \quad (9)$$

$$\delta_{\underline{V}-BR}^{\underline{V}} = \delta v (\cos \alpha \sin \delta, \sin \alpha \sin \delta, -\cos \delta)^T. \quad (10)$$

Let \underline{r}_{BR} be the equivalent conic planetocentric position of the spacecraft at release. Then by applying the subroutine STIMP consecutively to the states $(\underline{r}_{BR}, \underline{v}_{BR} + \delta_{\underline{U}-BR}^{\underline{V}})^T$ and $(\underline{r}_{BR}, \underline{v}_{BR} + \delta_{\underline{V}-BR}^{\underline{V}})^T$, the perturbed state vectors $\delta_{\underline{U}}^{\underline{\psi}}$ and $\delta_{\underline{V}}^{\underline{\psi}}$ can be generated. TPRTRG then approximates J as $(\delta_{\underline{U}}^{\underline{\psi}} | \delta_{\underline{V}}^{\underline{\psi}}) / \delta v$. Having approximated Δv_U and Δv_V , the roll release angle of miniprobe 1 and the tangential velocity at release are readily calculated from the formulae

$$v_T = \sqrt{\Delta v_U^2 + \Delta v_V^2} \quad (11)$$

$$\phi = \tan^{-1} (\Delta v_V / \Delta v_U). \quad (12)$$

It must be noted that ϕ represents the angle the velocity increment of the first miniprobe makes with the U axis. It should not be confused with the angle between the first probe arm and the U direction, which is $\pi/2$ radians less than ϕ .

Consider next the initial control estimate for virtual-mass minimum-miss controls. TPRTRG simply uses the minimum-miss conic control for this estimate. Hence, irrespective of the desired miniprobe propagation mode indicated by IPRØPI, the conic controls are first found. Then if only the conic controls are desired, a return is made to the calling program; otherwise the least-squares algorithm is repeated with virtual-mass rather than conic miniprobe propagation.

The third basic task of TPRTRG, namely using the subroutine GAUSLS to calculate the minimum-miss index controls, requires some explanation. First the four miniprobe release controls must be

identified. They are simply ϕ , v_T , δ_H , and α_H (see Fig. 1). Thus if \underline{u} denotes the control vector, then

$$\underline{u} = \begin{pmatrix} \phi \\ v_T \\ \delta_H \\ \alpha_H \end{pmatrix}. \quad (13)$$

By measuring the angles ϕ , α_H , and δ_H in radians and the velocity v_T in decameters/s all of the components of \underline{u} fall in the range from 0.1 to 10 as required by the subroutine GAUSLS when Jacobian matrices are generated by a uniform control perturbation of 10^{-5} .

Let $(B_i \cdot T_i)_A$ and $(B_i \cdot R_i)_A$ denote the actual B-plane pierce point coordinates of the i th miniprobe when the release control \underline{u} is applied. Let $(B_i \cdot T_i)_D$ and $(B_i \cdot R_i)_D$ represent the desired B-plane pierce point coordinates of the i th miniprobe based on the actual \underline{u} and the energy of that miniprobe. The six components of the constraint vector are then given as

$$\psi_i = C_i \left[(B_i \cdot T_i)_A - (B_i \cdot T_i)_D \right] \quad i = 1, 2, 3. \quad (14)$$

$$\psi_{i+3} = C_i \left[(B_i \cdot R_i)_A - (B_i \cdot R_i)_D \right] \quad (15)$$

Here the C_i 's are weighting factors input by the user to indicate the relative importance of achieving nearby impacts at the various miniprobe target sites. The subroutine TPPRØP is always used to calculate $\underline{\psi}$ given \underline{u} for whichever miniprobe propagation mode is specified by the flag IRPØP. Thus GAUSLS can be called to carry out the entire least-squares process of minimizing the miss-index $y = \underline{\psi}^T \underline{\psi}$ for either propagation mode once the initial control estimate \underline{u}_0 and the corresponding constraint $\underline{\psi}(\underline{u}_0)$ have been calculated. If the least-squares routine should fail to converge, the universal NOMNAL trouble flag, KWITT, is set to 1 to cause execution to terminate on return to the main program, and a return is made to the calling program, GIDANS.

The final responsibility of TPRTRG is to calculate the n-body miniprobe approach trajectory time histories and impact data for the bus and the miniprobes using the minimum-miss release controls corresponding to both the conic and virtual-mass propagation modes. The impact data for the bus are calculated in TPRTRG itself. Let \underline{r}_{BI} and $\underline{\rho}_{BI}$ denote the impact position vectors in the planetocentric ecliptic and probe-impact coordinate frames. The $\underline{\rho}_{BI}$ is calculated from \underline{r}_{BI} , which is available from the virtual-mass propagation, as

$$\underline{\rho}_{BI} = C \underline{r}_{BI}. \quad (16)$$

The right ascension, α_B , and declination, δ_B , of the bus impact site relative to the probe-impact frame are then readily calculated as

$$\alpha_B = \tan^{-1} \left(\rho_{BI} \right)_2 / \left(\rho_{BI} \right)_1 \quad (17)$$

$$\delta_B = \sin^{-1} \left(\rho_{BI} \right)_3. \quad (18)$$

$$\text{The flightpath angle, } \gamma_{BI} = \tan^{-1} \left(e r_I \sin \theta_I / p \right) \quad (19)$$

where r_I is the radius of the impact sphere, e is the eccentricity, p is the semilatus rectum and θ_I is the true anomaly at impact.

All of these conic elements refer to osculating hyperbola at impact. The magnitude of the bus impact velocity is calculated as

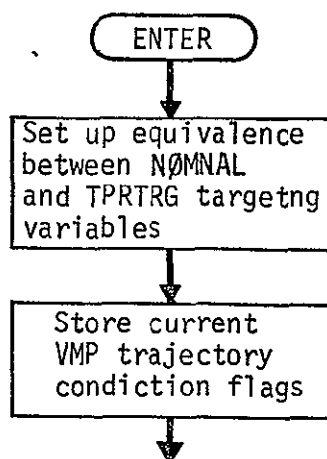
$$v_{BI} = \sqrt{u \left(2 / r_I - 1/a \right)} \quad (20)$$

where a is the semimajor axis of the osculating conic, and u is the gravitational constant of the planet.

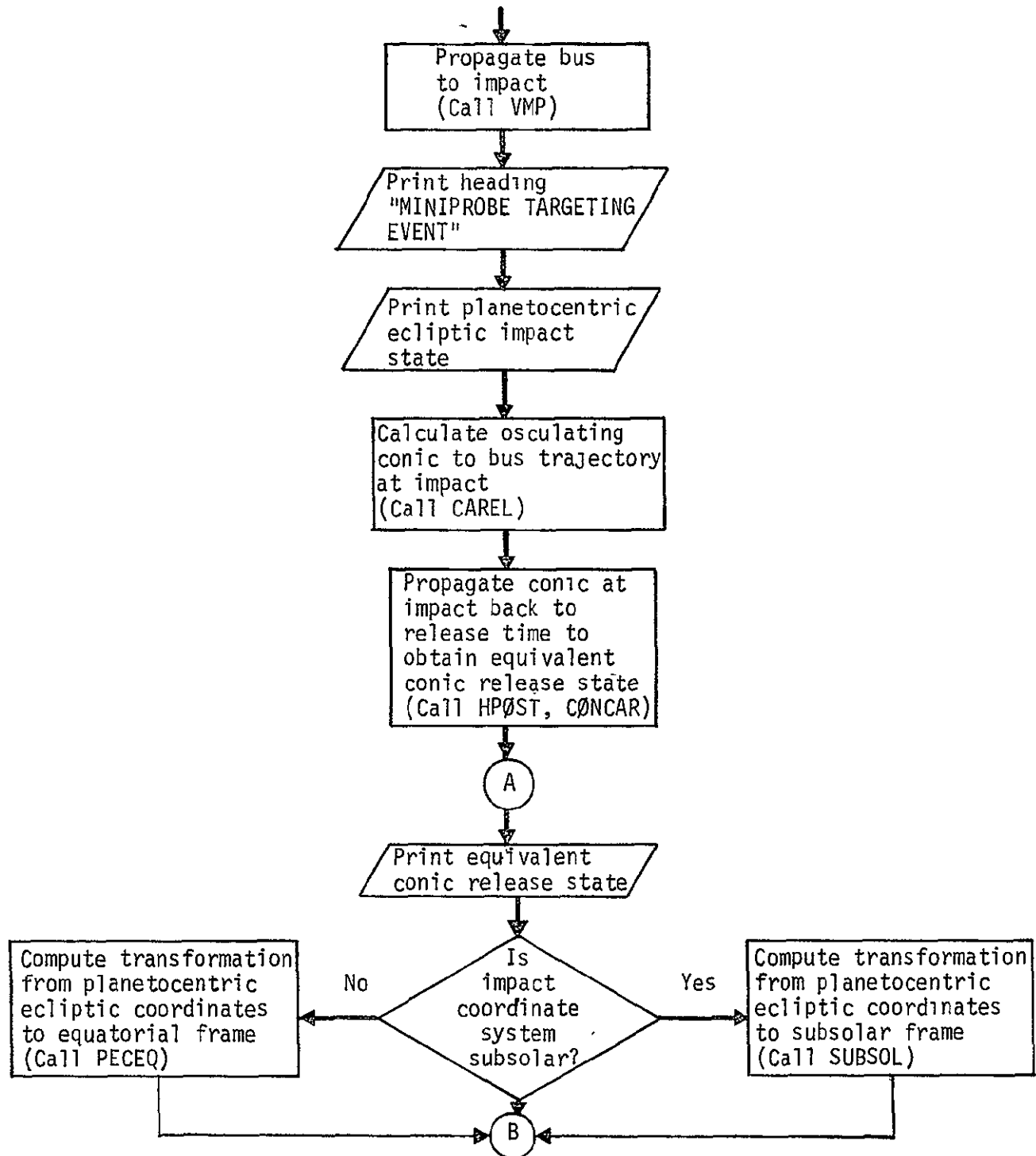
For the miniprobes, both the impact data and the virtual-mass approach trajectory time histories are generated by a call to TPRTRG with the least-squares status flag IFIN2 set to indicate that the miss-minimizing procedure is complete. When the least-squares algorithm is performed using conic miniprobe propagation, impact data are computed for both the conic and virtual-mass models.

The printout from TPRTRG is designed to meet two objectives: (1) to completely describe the minimum-miss miniprobe approach trajectories, and (2) to reveal any errors in the minimum-miss release controls caused by improper use of the program. To identify the type of nonlinear guidance event, the heading "Miniprobe Targeting Event" is printed first of all. Next pertinent data at release are printed. These consist of the planetocentric bus impact state and the equivalent conic release state as previously described. Then the printout from the miss-minimizing algorithm GAUSLS is provided in its entirety. After the minimum-miss release controls are found, they are printed out with a phrase indicating whether they correspond to the conic or virtual-mass model. If conic miniprobe propagation was used for the miss minimization, the conic-model probe impact data are printed. These include right ascension and declination of the impact point, together with time, velocity, and flightpath angle at impact for each of the miniprobes as well as the bus (numbered as probe number 0 in the printout). The bus data provided here are actually based on the initial virtual-mass propagation from the release state. The miniprobe data also contain the angles of attach, assuming the miniprobe longitudinal body axes remain parallel to the spacecraft spin axis at release. Next the time histories of the miniprobe minimum-miss approach trajectories are printed in succession from the subroutine VMP with print intervals of 5 days and 100 integration steps. Finally, the virtual-mass model probe impact information is printed. It is identical in content to the conic impact data except that the information is now based on virtual-mass propagation from the release state.

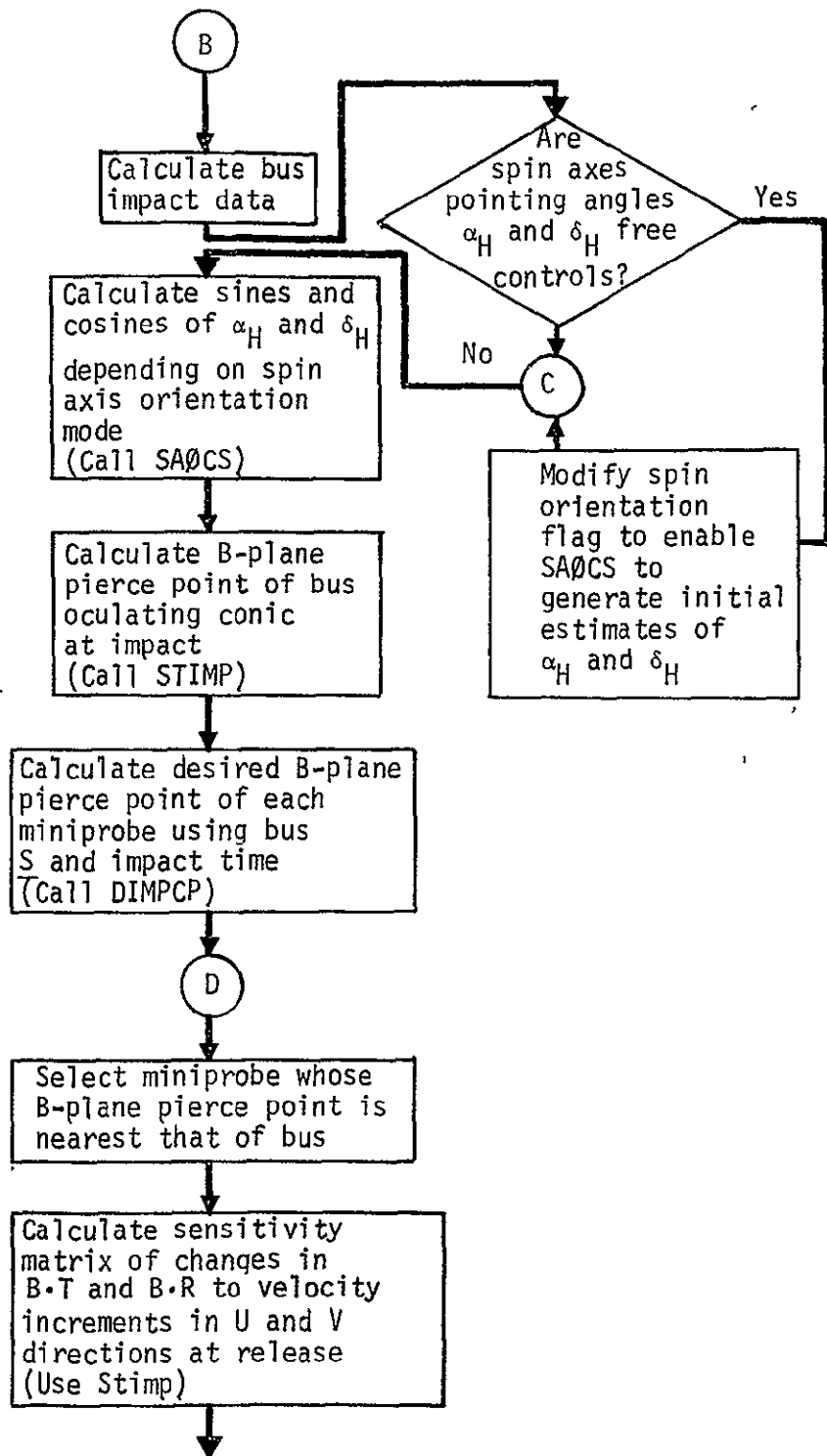
TPRTRG Flow Chart



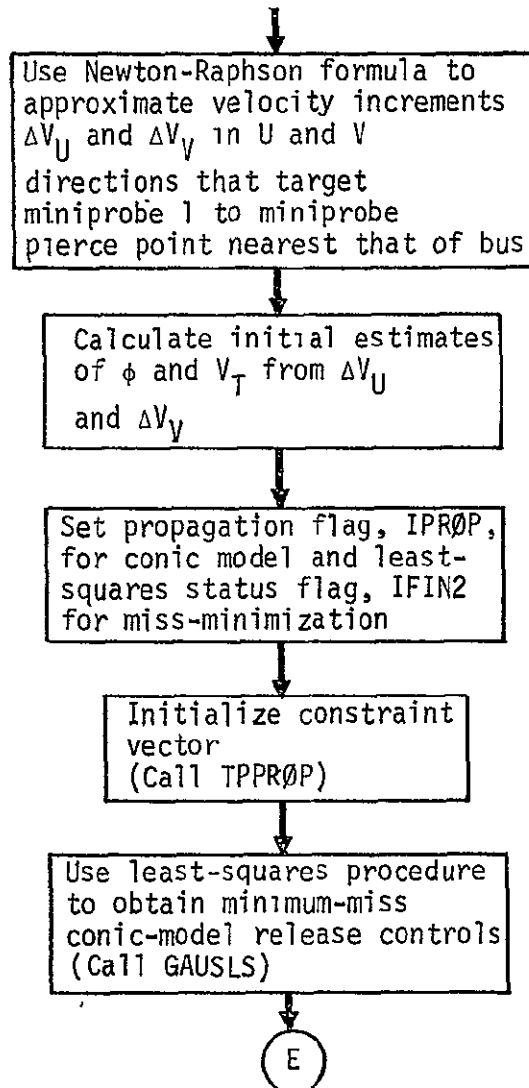
TPRTRG Flow Chart (cont)



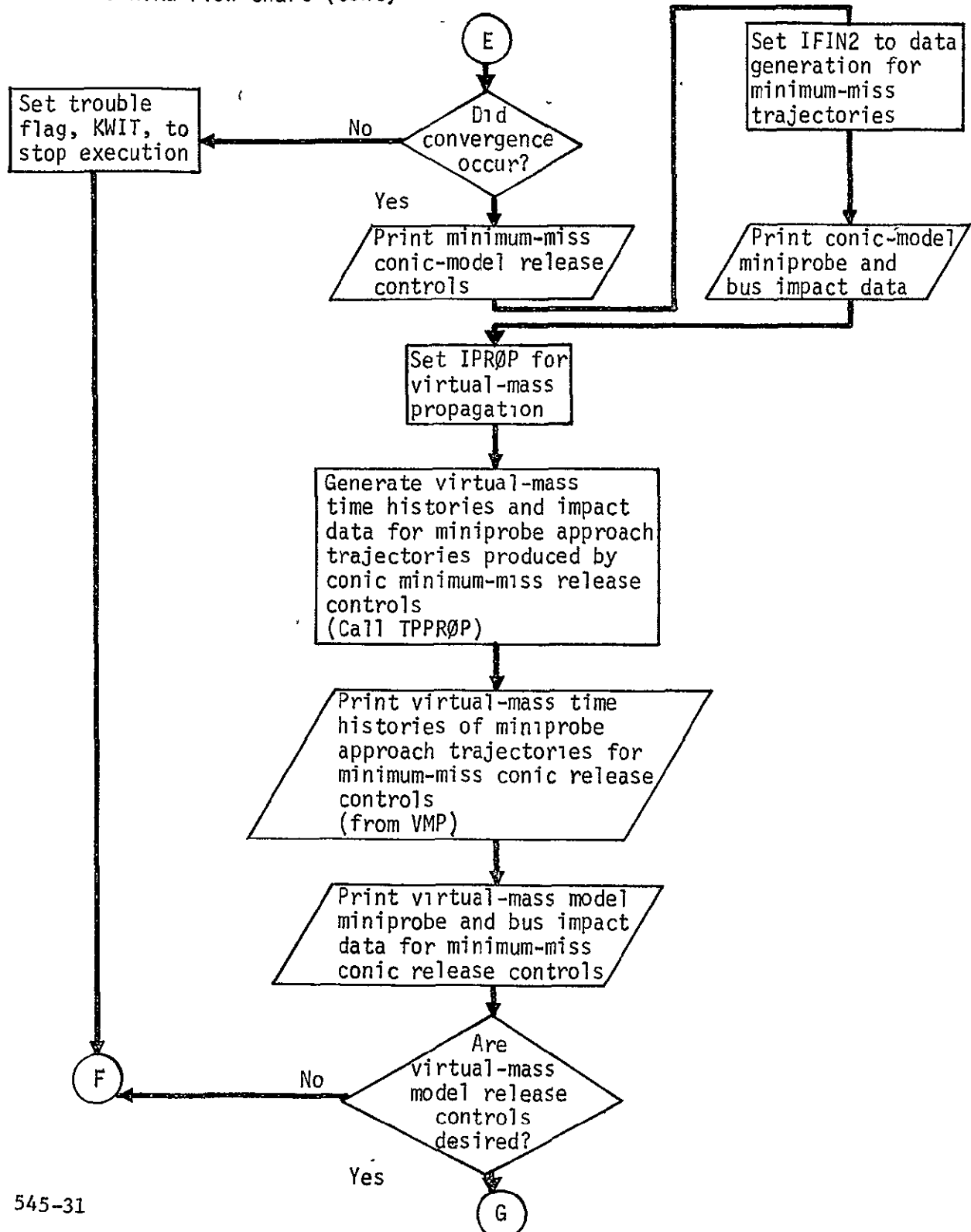
TPRTRG Flow Chart (cont)



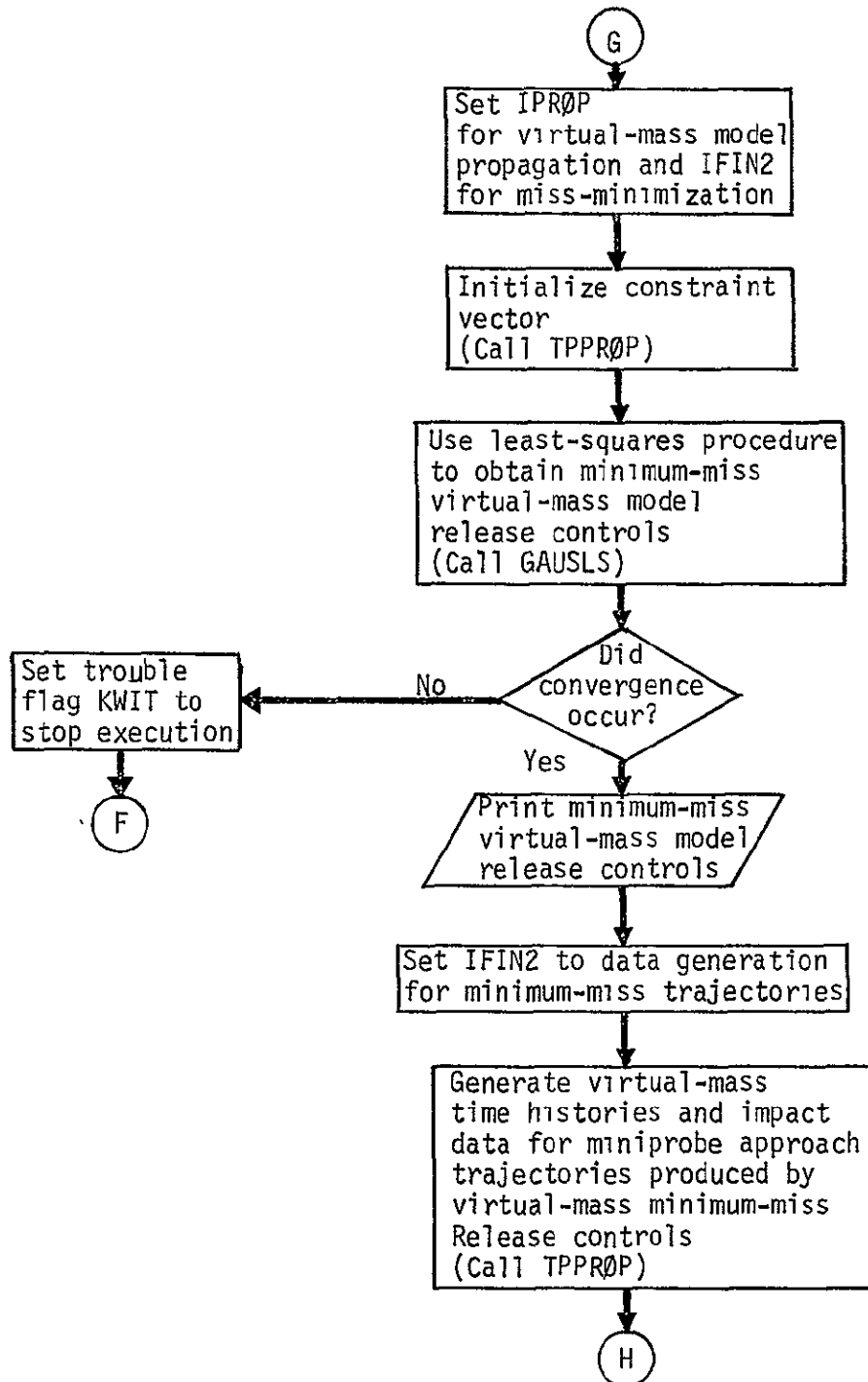
TPRTRG Flow Chart (cont)



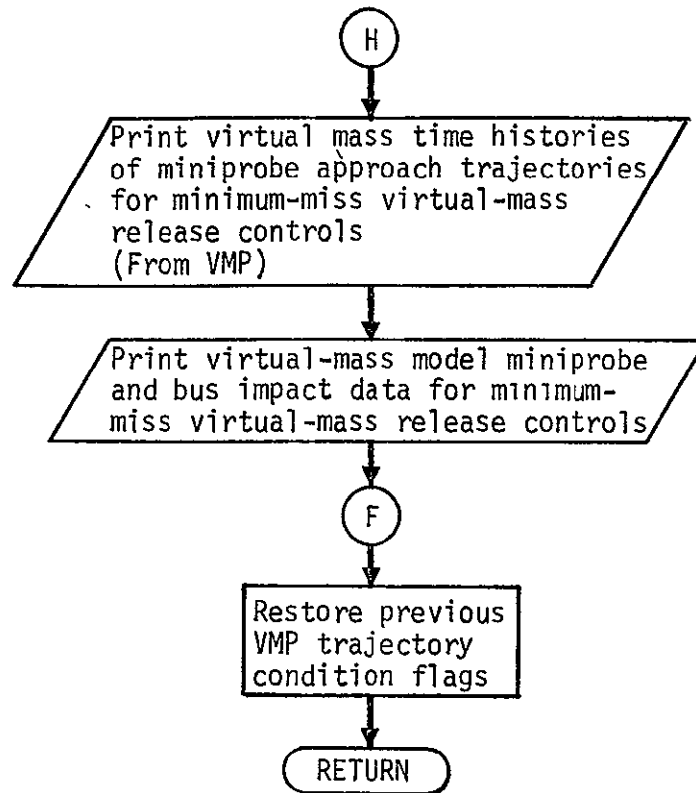
TPRTRG Flow Chart (cont)



TPRTRG Flow Chart (cont)



TPRTRG Flow Chart (concl)



Key to Symbols

α	- right ascension of spin axis
δ	- declination of spin axis
ϕ	- roll release angle of first miniprobe
\underline{v}_{T_i}	- tangential release velocity of i th miniprobe
\underline{H}	- spacecraft spin-axis unit vector
\underline{U}	- $\underline{H} \times \underline{Z}_{ec} / \underline{H} \times \underline{Z}_{ec} $
\underline{V}	- $\underline{H} \times \underline{U}$
$\underline{x}_{ec}, \underline{y}_{ec}, \underline{z}_{ec}$	- inertial ecliptic coordinate-axis unit vectors

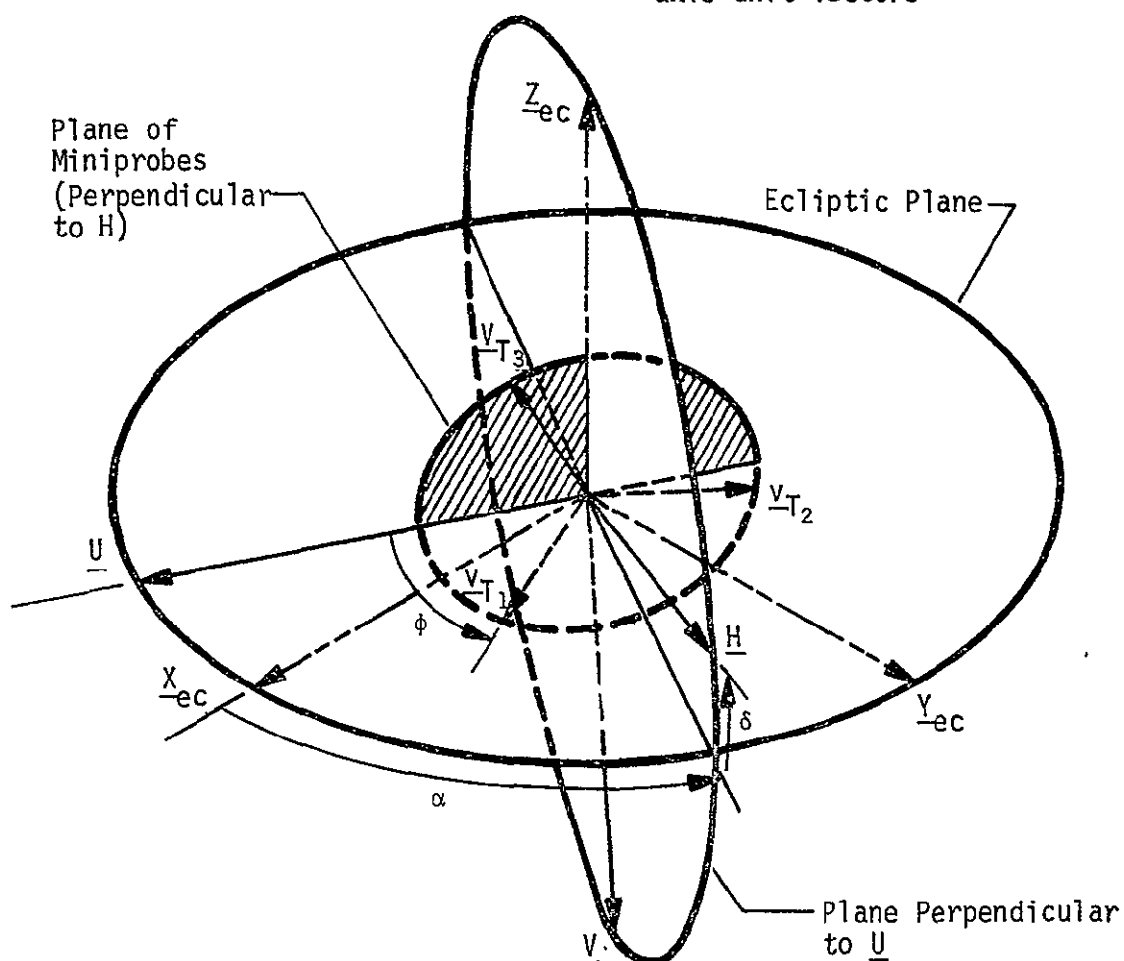


Figure 1 Miniprobe Release Geometry

SUBROUTINE TRAKM

PURPOSE: THE OBSERVATIONS AND OBSERVATION MATRIX FOR A GIVEN TYPE OF MEASUREMENT IS COMPUTED BY THIS ROUTINE

CALLING SEQUENCE: CALL TRAKM(HECV, ITRK, NR, IOBS, VECTOR)

ARGUMENT: HECV I POSITION AND VELOCITY OF SPACECRAFT AT TIME
OF MEASUREMENT

IOBS I CODE WHICH SPECIFIES IF MEASUREMENT OR
OBSERVATION MATRIX IS TO BE COMPUTED

```
ITRK      I  CODE WHICH SPECIFIES MEASUREMENT TYPE
           (CALLED MMCODE ELSEWHERE IN PROGRAM)
```

NR 0 NUMBER OF ROWS IN THE OBSERVATION MATRIX

VECT OR 0 ACTUAL MEASUREMENT

SUBROUTINES SUPPORTED: ERRANN PROBE

SUBROUTINES REQUIRED: EPHEM ORB STAPRL TARPRL

LOCAL SYMBOLS: AD1 INTERMEDIATE VARIABLE

AD2 INTERMEDIATE VARIABLE

AD3 INTERMEDIATE VARIABLE

A INTERMEDIATE VARIABLE

AL ALTITUDE

ALAT LATITUDE

ALON LONGITUDE

ALRAD INTERMEDIATE VARIABLE

A1 PARTIAL OF RANGE WITH RESPECT TO X

A2 PARTIAL OF RANGE WITH RESPECT TO Y

A3 PARTIAL OF RANGE WITH RESPECT TO Z

B1 PARTIAL OF RANGE-RATE WITH RESPECT TO X

B2 PARTIAL OF RANGE-RATE WITH RESPECT TO Y

B3 PARTIAL OF RANGE-RATE WITH RESPECT TO Z

CE COSINE OF OBLIQUITY OF EARTH

COAL	COSINE OF STAR-PLANET ANGLE
COSA	INTERMEDIATE VARIABLE
COSAZ	COSINE AZIMUTH
COSRA	COSINE RIGHT ASCENSION
CP	COSINE OF LONGITUDE + CONSTANT
DADP	PARTIALS OF STAR-PLANET ANGLE WITH RESPECT TO VEHICLE POSITION AND VELOCITY
DBDP	PARTIALS OF APPARENT PLANET DIAMETER WITH RESPECT TO VEHICLE POSITION AND VELOCITY
DD	INTERMEDIATE VARIABLE
DENOM	INTERMEDIATE VARIABLE
D	INTERMEDIATE TIME
EK	RANGE-RATE PARTIAL WITH RESPECT TO STATION LOCATION ERRORS
GECS	GEOCENTRIC EQUATORIAL COORDINATES OF STATION
GELS	GEOCENTRIC ECLIPTIC COORDINATES OF STATION
HECE	COORDINATES OF EARTH
HECP	COORDINATES OF TARGET PLANET
IA	TRACKING STATION LOCATION SELECTION CODE
ICD	CODE CORRESPONDING TO TRACKING STATION LOCATION ERRORS
IC	COLUMN NUMBER IN OBSERVATION MATRIX PARTITION WHERE EK IS TO BE STORED
IEND	VARIABLE INDEX VALUE
IR	STAR-PLANET ANGLE INDEX INCREMENT VALUE
NA	STAR-PLANET ANGLE INDEX LOWER LIMIT =1 FOR 3 STAR-PLANET ANGLES =ITRK-10 FOR SINGLE STAR-PLANET ANGLES
NC	STAR-PLANET ANGLE INDEX UPPER LIMIT =3 FOR 3 STAR-PLANET ANGLES

=ITRK-10 FOR SINGLE STAR-PLANET ANGLES

PAR	PARTIALS RETURNED FROM SUBROUTINE TARPRL
PAT1	INTERMEDIATE VARIABLE
PAT2	INTERMEDIATE VARIABLE
PA	PARTIALS
P2	INTERMEDIATE VARIABLE
RA	RIGHT ASCENSION
RAONTP	RADIUS OF TARGET PLANET
RAS	INTERMEDIATE VARIABLE
RHOP	INTERMEDIATE VECTOR
RRATE	RANGE-RATE
R1	RANGE
R2	SQUARE OF RANGE
SA	PARTIALS OF STAR-PLANET ANGLES WITH RESPECT TO VEHICLE POSITION
SE	SINE OF OBLIQUITY OF EARTH
SIAL	SINE OF STAR-PLANET ANGLE
SINAZ	SINE AZIMUTH
SIND	INTERMEDIATE VARIABLE
SINRA	SINE RIGHT ASCENSION
SP	SINE OF LONGITUDE + CONSTANT
SUM	INTERMEDIATE VARIABLE
SUM1	INTERMEDIATE VARIABLE
VEC	INTERMEDIATE VECTOR
ZZ1	INTERMEDIATE VARIABLE
ZZ2	INTERMEDIATE VARIABLE

COMMON COMPUTED/USED:	AAL	AM	H	NO	T
-----------------------	-----	----	---	----	---

	XP	AN			
COMMON COMPUTED:	G				
COMMON USED:	ALNGTH	DATEJ	DELT	EM3	EPS
	F	IAUGDC	IAUGIN	IAUGMC	IAUG
	IBARY	NBOD	NB	NTP	OMEGA
	ONE	RADIUS	SAL	SLAT	SLON
	TM	TRTM1	TWO	UNIVT	UST
	VST	WST	ZERO		

TRAKM Analysis

The linearized observation equation can be written as

$$y = Hx + Mx_s + Gu + Lv + Nw$$

where y is the observable, x is the spacecraft state, and x_s , u , v , and w are solve-for, dynamic consider, measurement consider, and ignore parameter vectors, respectively. The function of subroutine TRAKM is to compute the observation matrix partitions H , M , G , L , and N , which indicate the sensitivity of the observable y to changes in x , x_s , u , v , and w , respectively, in the error analysis/generalized covariance analysis program. The matrix N is computed only for a generalized covariance analysis.

Except for computation of the ignore parameter observation matrix partition N , TRAKM is equivalent to subroutine TRAKS, which is used in the simulation program. See subroutine TRAKS for further analytical details and a flow chart.

SUBROUTINE TRAKS

PURPOSE: TO COMPUTE ALL OBSERVATION MATRIX PARTITIONS FOR THE MEASUREMENT TYPE AND TO COMPUTE THE MEASUREMENT ITSELF.

CALLING SEQUENCE: CALL TRAKS(HECV,ITRK,NR,I OBS,VECTOR)

ARGUMENT: HECV I POSITION AND VELOCITY OF SPACECRAFT AT TIME OF MEASUREMENT

I OBS I CODE WHICH SPECIFIES IF MEASUREMENT OR OBSERVATION MATRIX IS TO BE COMPUTED

ITRK I CODE WHICH SPECIFIES MEASUREMENT TYPE (CALLED MMCODE ELSEWHERE IN PROGRAM)

NR O NUMBER OF ROWS IN THE OBSERVATION MATRIX

VECTOR O ACTUAL MEASUREMENT

SUBROUTINES SUPPORTED: SIMULL

SUBROUTINES REQUIRED: EPHEM ORB STAPRL TARPRL

LOCAL SYMBOLS: AD1 INTERMEDIATE VARIABLE

AD2 INTERMEDIATE VARIABLE

AD3 INTERMEDIATE VARIABLE

A INTERMEDIATE VARIABLE

ALAT LATITUDE

ALON LONGITUDE

AL ALTITUDE

ALRAD INTERMEDIATE VARIABLE

A1 PARTIAL OF RANGE WITH RESPECT TO X

A2 PARTIAL OF RANGE WITH RESPECT TO Y

A3 PARTIAL OF RANGE WITH RESPECT TO Z

B1 PARTIAL OF RANGE-RATE WITH RESPECT TO X

B2 PARTIAL OF RANGE-RATE WITH RESPECT TO Y

B3 PARTIAL OF RANGE-RATE WITH RESPECT TO Z

GE COSINE OF OBLIQUITY OF EARTH

COAL	COSINE OF STAR-PLANET ANGLE
COSA	INTERMEDIATE VARIABLE
COSAZ	COSINE AZIMUTH
COSRA	COSINE RIGHT ASCENSION
CP	COSINE OF LONGITUDE + CONSTANT
DADP	PARTIALS OF STAR-PLANET ANGLE WITH RESPECT TO VEHICLE POSITION AND VELOCITY
DBDP	PARTIALS OF APPARENT PLANET DIAMETER WITH RESPECT TO VEHICLE POSITION AND VELOCITY
DD	INTERMEDIATE VARIABLE
DENOM	INTERMEDIATE VARIABLE
D	INTERMEDIATE TIME
EK	RANGE-RATE PARTIAL WITH RESPECT TO STATION LOCATION ERRORS
GECS	GEOCENTRIC EQUATORIAL COORDINATES OF STATION
GELS	GEOCENTRIC ECLIPTIC COORDINATES OF STATION
HECE	COORDINATES OF EARTH
HECP	COORDINATES OF TARGET PLANET
IA	TRACKING STATION LOCATION SELECTION CODE
ICD	CODE CORRESPONDING TO TRACKING STATION LOCATION ERRORS
IC	COLUMN NUMBER IN OBSERVATION MATRIX PARTITION WHERE EK IS TO BE STORED
IEND	VARIABLE INDEX VALUE
IR	STAR-PLANET ANGLE INDEX INCREMENT VALUE
NA	STAR-PLANET ANGLE INDEX LOWER LIMIT =1 FOR 3 STAR-PLANET ANGLES =ITRK-10 FOR SINGLE STAR-PLANET ANGLES
NC	STAR-PLANET ANGLE INDEX UPPER LIMIT =3 FOR 3 STAR-PLANET ANGLES

=ITRK-10 FOR SINGLE STAR-PLANET ANGLES

PAR	PARTIALS RETURNED FROM SUBROUTINE TARPRL
PAT1	INTERMEDIATE VARIABLE
PAT2	INTERMEDIATE VARIABLE
PA	PARTIALS
P2	INTERMEDIATE VARIABLE
RA	RIGHT ASCENSION
RADNTP	RADIUS OF TARGET PLANET
RAS	INTERMEDIATE VARIABLE
RRATE	RANGE-RATE
R1	RANGE
R2	SQUARE OF RANGE
SA	PARTIALS OF STAR-PLANET ANGLES WITH RESPECT TO VEHICLE POSITION
SE	SINE OF OBLIQUITY OF EARTH
SIAL	SINE OF STAR-PLANET ANGLE
SINAZ	SINE AZIMUTH
SIND	INTERMEDIATE VARIABLE
SINRA	SINE RIGHT ASCENSION
SP	SINE OF LONGITUDE + CONSTANT
SUM	INTERMEDIATE VARIABLE
SUM1	INTERMEDIATE VARIABLE
VEC	INTERMEDIATE VECTOR
ZZ1	INTERMEDIATE VARIABLE
ZZ2	INTERMEDIATE VARIABLE

COMMON COMPUTED/USED:	AAL	AM	H	NO	XP
-----------------------	-----	----	---	----	----

COMMON COMPUTED:

G

COMMON USED:

ALNGTH	DATEJ	DELT	EM3	EPS
F	IAUGDC	IAUGIN	IAUGMC	IAUG
IBARY	N80D	NB	NTP	OMEGA
ONE	RADIUS	SAL	SLAT	SLB
SLON	TM	TRTMB	TRTM1	TWO
UNIVT	UST	VST	WST	ZERO

TRAKS Analysis

Subroutine TRAKS performs two functions in the simulation mode. The first function, which corresponds to $I\emptyset BS = 0$, is to compute all observation matrix partitions for the measurement type indicated by ITRK. The second function, which corresponds to $I\emptyset BS \neq 0$, is to compute the measurement itself. If $I\emptyset BS = 1$, TRAKS computes the measurement corresponding to the most recent nominal spacecraft state. If $I\emptyset BS = 2$, TRAKS computes the measurement corresponding to the actual spacecraft state, and, if the measurement is a range or range-rate measurement, to the actual tracking station locations. The number of rows, NR, in the measurement and the observation matrix partitions is also computed.

A general measurement has form

$$\vec{Y} = \vec{Y}(\vec{X}, \vec{p}, t)$$

where \vec{X} is the spacecraft position/velocity state at time t and \vec{p} is a vector of parameters. This equation can be linearized about nominal \vec{X} and \vec{p} to obtain

$$\delta \vec{Y} = \left(\frac{\partial \vec{Y}}{\partial \vec{X}} \right)^* \delta \vec{X} + \left(\frac{\partial \vec{Y}}{\partial \vec{p}} \right)^* \delta \vec{p}$$

where $()^*$ indicates matrices are evaluated at the nominal condition. This perturbation equation can be rewritten as

$$\delta \vec{Y} = H \delta \vec{X} + M \delta \vec{x}_s + G \delta \vec{u} + L \delta \vec{v}$$

where $H = \left(\frac{\partial \vec{Y}}{\partial \vec{X}} \right)^*$, and $\left(\frac{\partial \vec{Y}}{\partial \vec{p}} \right)^*$ is distributed among the M , G , and L partitions to correspond to the partition of the parameter vector $\delta \vec{p}$ into solve-for parameters $\delta \vec{x}_s$, dynamic consider parameters $\delta \vec{u}$, and measurement consider parameters $\delta \vec{v}$.

In the remainder of this section the measurement equation and all partial derivatives required to construct the H , M , G , and L observation matrix partitions will be summarized for each measurement type.

A. Range measurement ρ .

A range measurement has form

$$\rho = \rho(\vec{X}, R, \theta, \phi, t)$$

where R , θ , and ϕ are the radius, latitude, and longitude of the relevant tracking station.

More explicitly,

$$\rho = \left[(X - X_E - X_S)^2 + (Y - Y_E - Y_S)^2 + (Z - Z_E - Z_S)^2 \right]^{\frac{1}{2}}$$

where X, Y, Z = inertial position components of spacecraft
 X_E, Y_E, Z_E = inertial position components of Earth
 X_S, Y_S, Z_S = station position components relative to Earth.

X_S, Y_S , and Z_S are related to R, θ , and ϕ as follows:

$$X_S = R \cos \theta \cos G$$

$$Y_S = R \cos \theta \cos \epsilon \sin G + R \sin \theta \sin \epsilon$$

$$Z_S = -R \cos \theta \sin \epsilon \sin G + R \sin \theta \cos \epsilon$$

where ϵ is the obliquity of the Earth, and

$$G = \phi + \text{GHA}$$

where GHA is the Greenwich hour angle at time t .

Partials of ρ with respect to spacecraft state are given by

$$\frac{\partial \rho}{\partial X} = \frac{1}{\rho} (X - X_E - X_S) \quad \frac{\partial \rho}{\partial \dot{X}} = 0$$

$$\frac{\partial \rho}{\partial Y} = \frac{1}{\rho} (Y - Y_E - Y_S) \quad \frac{\partial \rho}{\partial \dot{Y}} = 0$$

$$\frac{\partial \rho}{\partial Z} = \frac{1}{\rho} (Z - Z_E - Z_S) \quad \frac{\partial \rho}{\partial \dot{Z}} = 0$$

Partials of ρ with respect to R, θ , and ϕ are given by

$$\frac{\partial \rho}{\partial R} = \frac{\partial \rho}{\partial X_S} \cdot \frac{\partial X_S}{\partial R} + \frac{\partial \rho}{\partial Y_S} \cdot \frac{\partial Y_S}{\partial R} + \frac{\partial \rho}{\partial Z_S} \cdot \frac{\partial Z_S}{\partial R}$$

$$\frac{\partial \rho}{\partial \theta} = \frac{\partial \rho}{\partial X_S} \cdot \frac{\partial X_S}{\partial \theta} + \frac{\partial \rho}{\partial Y_S} \cdot \frac{\partial Y_S}{\partial \theta} + \frac{\partial \rho}{\partial Z_S} \cdot \frac{\partial Z_S}{\partial \theta}$$

$$\frac{\partial \rho}{\partial \phi} = \frac{\partial \rho}{\partial X_S} \cdot \frac{\partial X_S}{\partial \phi} + \frac{\partial \rho}{\partial Y_S} \cdot \frac{\partial Y_S}{\partial \phi} + \frac{\partial \rho}{\partial Z_S} \cdot \frac{\partial Z_S}{\partial \phi}$$

where

$$\frac{\partial \rho}{\partial X_S} = - \frac{\partial \rho}{\partial X}, \quad \frac{\partial \rho}{\partial Y_S} = - \frac{\partial \rho}{\partial Y}, \quad \frac{\partial \rho}{\partial Z_S} = - \frac{\partial \rho}{\partial Z}$$

and the negatives of the partials of X_S , Y_S , and Z_S with respect to R , θ , and ϕ are summarized in the subroutine STAPRL analysis.

B. Range-rate measurement $\dot{\rho}$.

A range-rate measurement has form

$$\dot{\rho} = \dot{\rho}(\vec{X}, R, \theta, \phi, t)$$

where all arguments have been defined previously. More explicitly,

$$\dot{\rho} = \frac{\rho_x \dot{\rho}_x + \rho_y \dot{\rho}_y + \rho_z \dot{\rho}_z}{\rho}$$

$$\begin{aligned} \text{where} \quad \rho_x &= X - X_E - X_S & \dot{\rho}_x &= \dot{X} - \dot{X}_E - \dot{X}_S \\ \rho_y &= Y - Y_E - Y_S & \dot{\rho}_y &= \dot{Y} - \dot{Y}_E - \dot{Y}_S \\ \rho_z &= Z - Z_E - Z_S & \dot{\rho}_z &= \dot{Z} - \dot{Z}_E - \dot{Z}_S \end{aligned}$$

\dot{X}_S , \dot{Y}_S , and \dot{Z}_S are related to R , θ , and ϕ as follows:

$$\begin{aligned} \dot{X}_S &= -\omega R \cos \theta \sin \phi \\ \dot{Y}_S &= \omega R \cos \theta \cos \phi \cos G \\ \dot{Z}_S &= -\omega R \cos \theta \sin \phi \cos G \end{aligned}$$

where ω is the rotational rate of the Earth.

Partial of $\dot{\rho}$ with respect to spacecraft state are given by

$$\begin{aligned} \frac{\partial \dot{\rho}}{\partial X} &= \frac{\dot{\rho}_x}{\rho} - \frac{\rho_x \dot{\rho}}{\rho^2} & \frac{\partial \dot{\rho}}{\partial \dot{X}} &= \frac{\rho_x}{\rho} \\ \frac{\partial \dot{\rho}}{\partial Y} &= \frac{\dot{\rho}_y}{\rho} - \frac{\rho_y \dot{\rho}}{\rho^2} & \frac{\partial \dot{\rho}}{\partial \dot{Y}} &= \frac{\rho_y}{\rho} \\ \frac{\partial \dot{\rho}}{\partial Z} &= \frac{\dot{\rho}_z}{\rho} - \frac{\rho_z \dot{\rho}}{\rho^2} & \frac{\partial \dot{\rho}}{\partial \dot{Z}} &= \frac{\rho_z}{\rho} \end{aligned}$$

The partial of $\dot{\rho}$ with respect to R is given by

$$\begin{aligned} \frac{\partial \dot{\rho}}{\partial R} = & \frac{\partial \dot{\rho}}{\partial X_S} \cdot \frac{\partial X_S}{\partial R} + \frac{\partial \dot{\rho}}{\partial Y_S} \cdot \frac{\partial Y_S}{\partial R} + \frac{\partial \dot{\rho}}{\partial Z_S} \cdot \frac{\partial Z_S}{\partial R} + \\ & \frac{\partial \dot{\rho}}{\partial \dot{X}_S} \cdot \frac{\partial \dot{X}_S}{\partial R} + \frac{\partial \dot{\rho}}{\partial \dot{Y}_S} \cdot \frac{\partial \dot{Y}_S}{\partial R} + \frac{\partial \dot{\rho}}{\partial \dot{Z}_S} \cdot \frac{\partial \dot{Z}_S}{\partial R} \end{aligned}$$

where

$$\frac{\partial \dot{\rho}}{\partial X_S} = - \frac{\partial \dot{\rho}}{\partial X}, \text{ etc.}$$

$$\text{and } \frac{\partial \dot{\rho}}{\partial \dot{X}_S} = - \frac{\partial \dot{\rho}}{\partial \dot{X}}, \text{ etc.}$$

The negatives of the partials of X_S , Y_S , Z_S , \dot{X}_S , \dot{Y}_S , and \dot{Z}_S with respect to R , θ , and ϕ are summarized in the subroutine 'STAPRL' analysis. Partial of $\dot{\rho}$ with respect to θ and ϕ are treated similarly.

C. Star-planet angle measurement α .

A star-planet angle measurement has form α

$$\alpha = \alpha(\bar{X}, a, e, i, \Omega, \omega, M)$$

where a , e , i , Ω , ω , and M are the standard set of target planet orbital elements.

If we define $\vec{\rho} = (\rho_x, \rho_y, \rho_z)$ to be the position of the target planet relative to the spacecraft and (u, v, w) to be the direction cosines of the relevant star, then

$$\alpha = \cos^{-1} \left[\frac{1}{\rho} (u\rho_x + v\rho_y + w\rho_z) \right]$$

where

$$\rho_x = X_p - X, \quad \rho_y = Y_p - Y, \quad \rho_z = Z_p - Z,$$

and (X_p, Y_p, Z_p) represent the position coordinates of the target planet.

Partials of α with respect to spacecraft state are given by

$$\begin{aligned}\frac{\partial \alpha}{\partial X} &= \frac{1}{\sin \alpha} \left(\frac{u}{\rho} - \frac{\rho_x \cos \alpha}{\rho^2} \right) & \frac{\partial \alpha}{\partial \dot{X}} &= 0 \\ \frac{\partial \alpha}{\partial Y} &= \frac{1}{\sin \alpha} \left(\frac{v}{\rho} - \frac{\rho_y \cos \alpha}{\rho^2} \right) & \frac{\partial \alpha}{\partial \dot{Y}} &= 0 \\ \frac{\partial \alpha}{\partial Z} &= \frac{1}{\sin \alpha} \left(\frac{w}{\rho} - \frac{\rho_z \cos \alpha}{\rho^2} \right) & \frac{\partial \alpha}{\partial \dot{Z}} &= 0\end{aligned}$$

where

$$\sin \alpha = + \left[1 - \cos^2 \alpha \right]^{\frac{1}{2}}.$$

The partial of α with respect to target planet semi-major axis is given by

$$\frac{\partial \alpha}{\partial a} = \frac{\partial \alpha}{\partial X_p} \cdot \frac{\partial X_p}{\partial a} + \frac{\partial \alpha}{\partial Y_p} \cdot \frac{\partial Y_p}{\partial a} + \frac{\partial \alpha}{\partial Z_p} \cdot \frac{\partial Z_p}{\partial a}$$

$$\text{where } \frac{\partial \alpha}{\partial X_p} = - \frac{\partial \alpha}{\partial X}, \quad \frac{\partial \alpha}{\partial Y_p} = - \frac{\partial \alpha}{\partial Y}, \quad \frac{\partial \alpha}{\partial Z_p} = - \frac{\partial \alpha}{\partial Z},$$

and partials of X_p , Y_p , and Z_p with respect to semi-major axis are summarized in the subroutine TARPRL analysis. Partial of α with respect to \dot{X}_p , \dot{Y}_p , and \dot{Z}_p do not appear in the above expression since they are all zero. Partial of α with respect to the remaining target planet orbital elements are treated similarly.

D. Apparent planet diameter measurement β .

An apparent planet diameter measurement has form

$$\beta = \beta(\vec{X}, a, e, i, \Omega, \omega, M)$$

where all arguments have been defined previously.

Defining $\vec{\rho} = (\rho_x, \rho_y, \rho_z)$ to be the position of the target planet relative to the spacecraft and R_p to be the radius of the target planet, the apparent planet diameter can then be written as

$$\beta = 2 \sin^{-1} \left(\frac{R_p}{\rho} \right)$$

Partials of β with respect to spacecraft state are given by

$$\frac{\partial \beta}{\partial X} = \frac{2 R_p \rho_x}{\rho^2 [\rho^2 - R_p^2]^{\frac{1}{2}}} \quad \frac{\partial \beta}{\partial \dot{X}} = 0$$

$$\frac{\partial \beta}{\partial Y} = \frac{2 R_p \rho_y}{\rho^2 [\rho^2 - R_p^2]^{\frac{1}{2}}} \quad \frac{\partial \beta}{\partial \dot{Y}} = 0$$

$$\frac{\partial \beta}{\partial Z} = \frac{2 R_p \rho_z}{\rho^2 [\rho^2 - R_p^2]^{\frac{1}{2}}} \quad \frac{\partial \beta}{\partial \dot{Z}} = 0$$

The partial of β with respect to target planet semi-major axis is given by

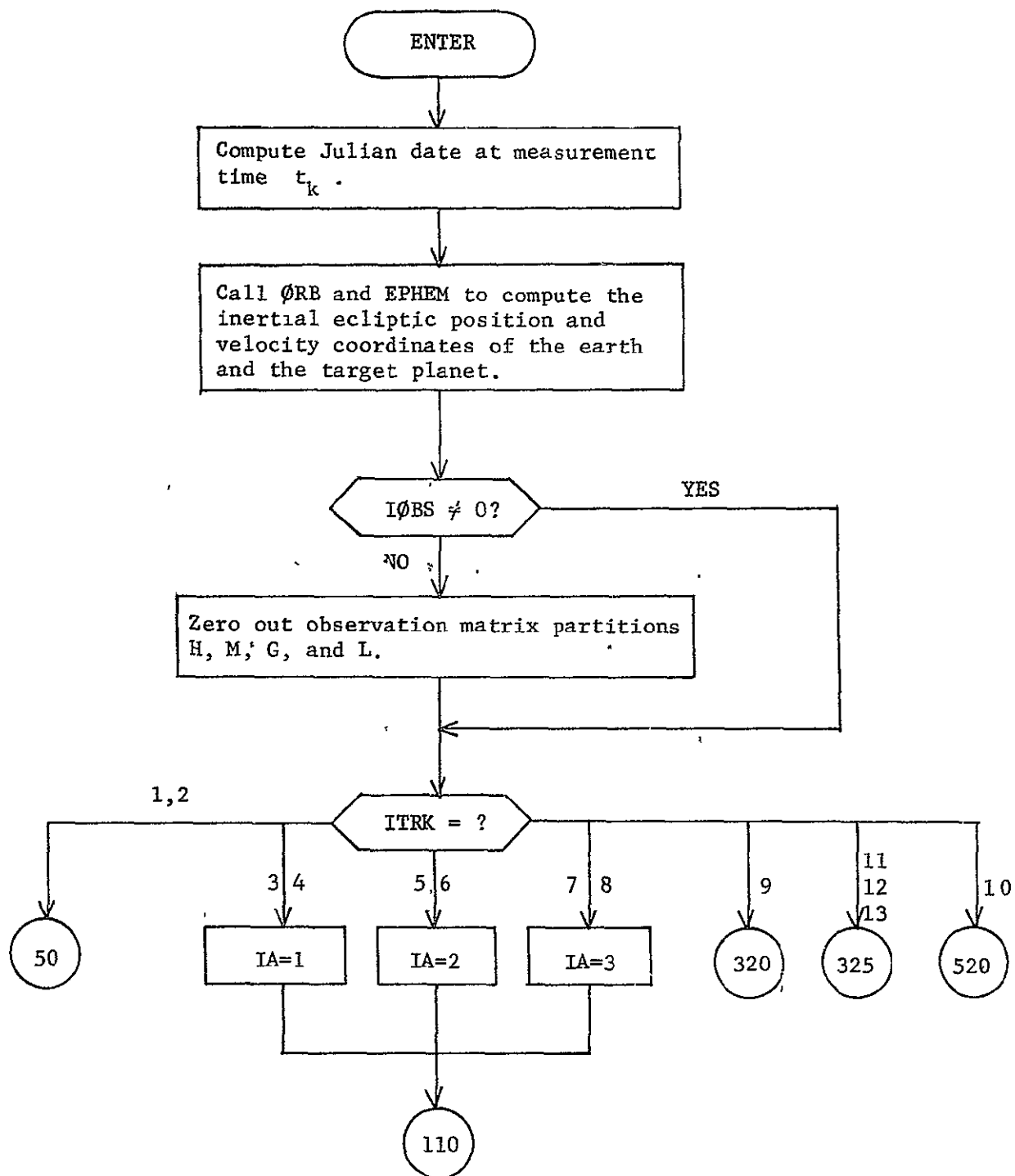
$$\frac{\partial \beta}{\partial a} = \frac{\partial \beta}{\partial X_p} \cdot \frac{\partial X_p}{\partial a} + \frac{\partial \beta}{\partial Y_p} \cdot \frac{\partial Y_p}{\partial a} + \frac{\partial \beta}{\partial Z_p} \cdot \frac{\partial Z_p}{\partial a}$$

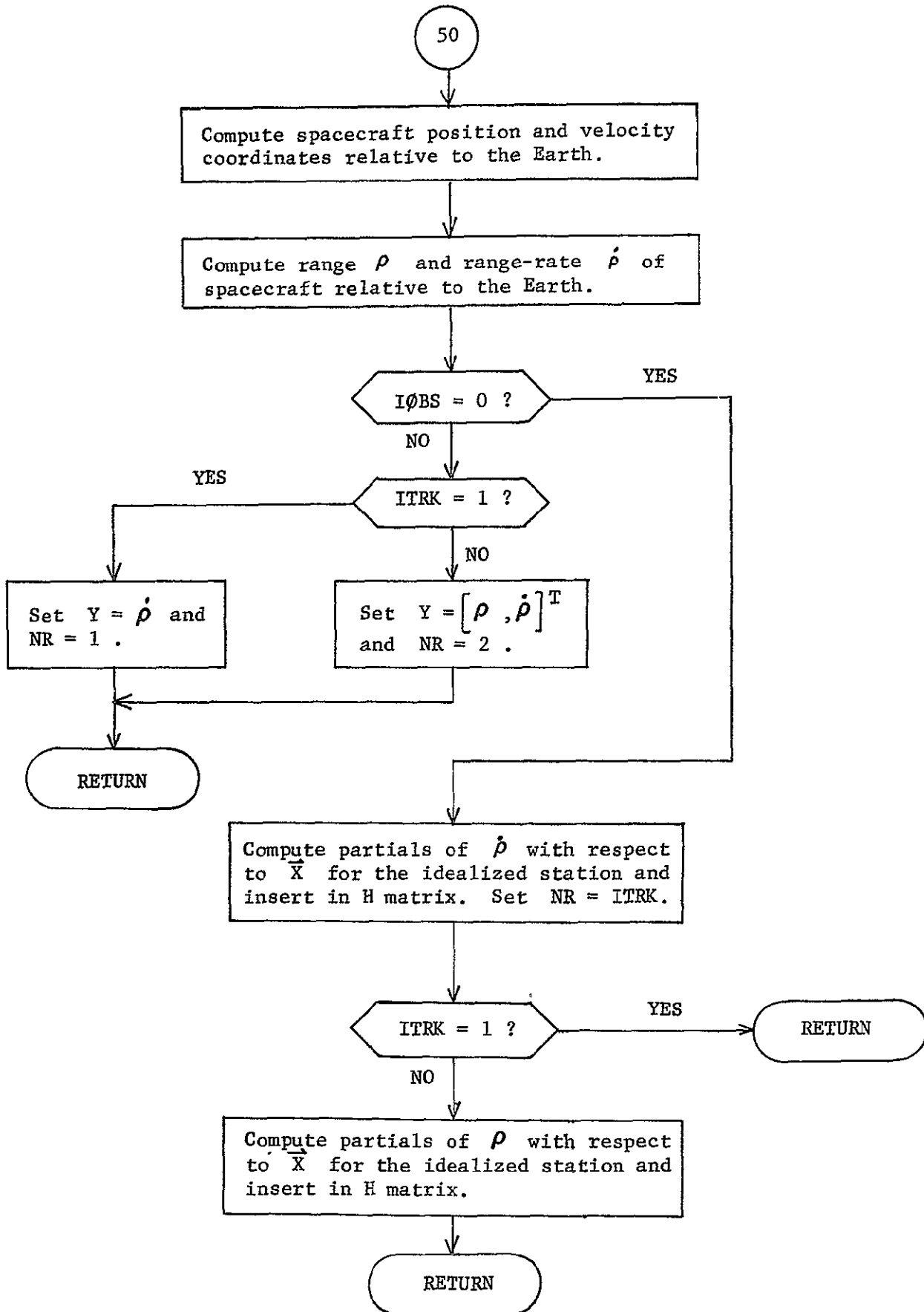
$$\text{where } \frac{\partial \beta}{\partial X_p} = - \frac{\partial \beta}{\partial X}, \quad \frac{\partial \beta}{\partial Y_p} = - \frac{\partial \beta}{\partial Y}, \quad \frac{\partial \beta}{\partial Z_p} = - \frac{\partial \beta}{\partial Z},$$

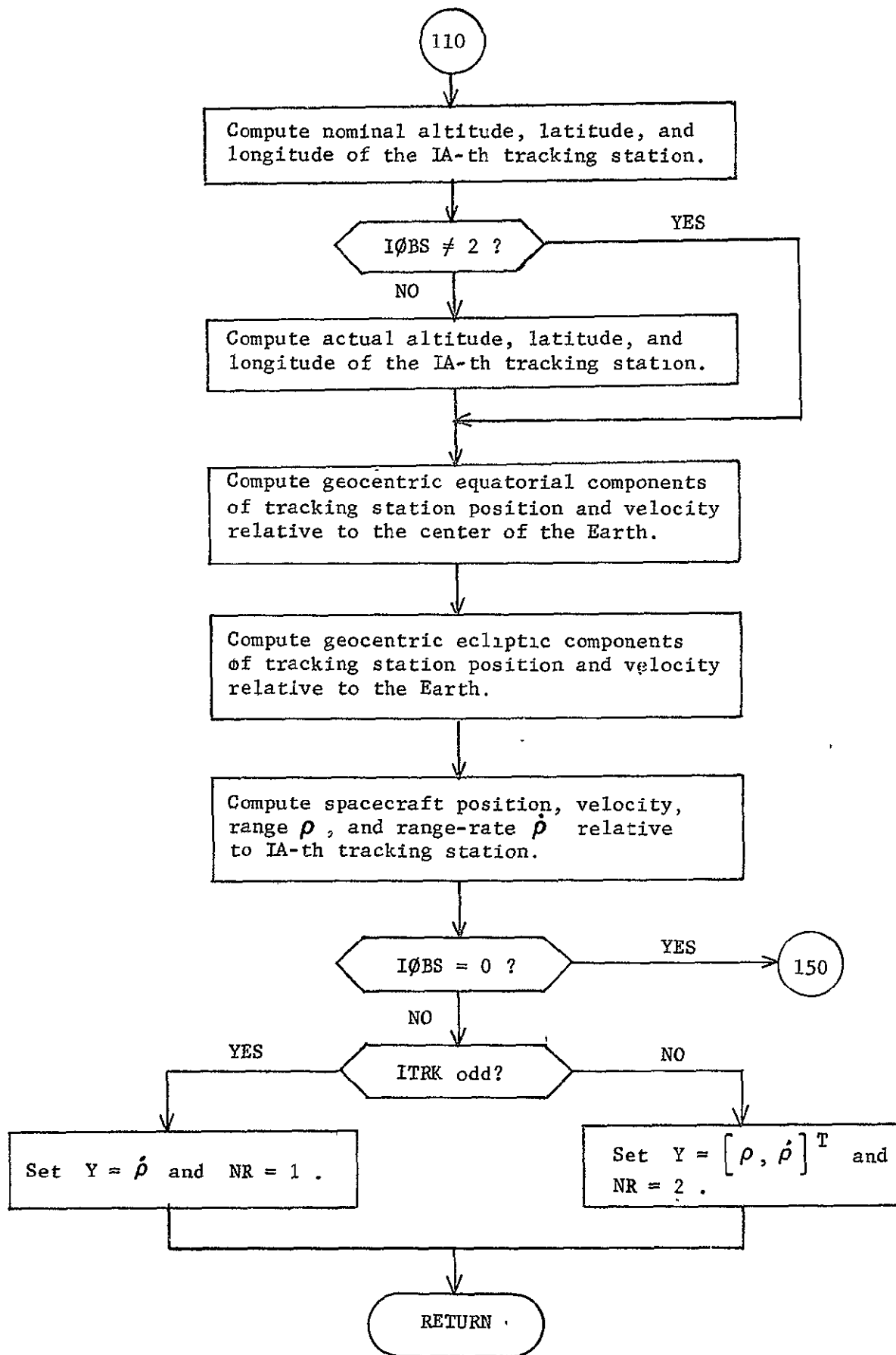
and partials of X_p , Y_p , and Z_p with respect to semi-major axis are summarized in the subroutine TARPRL analysis.

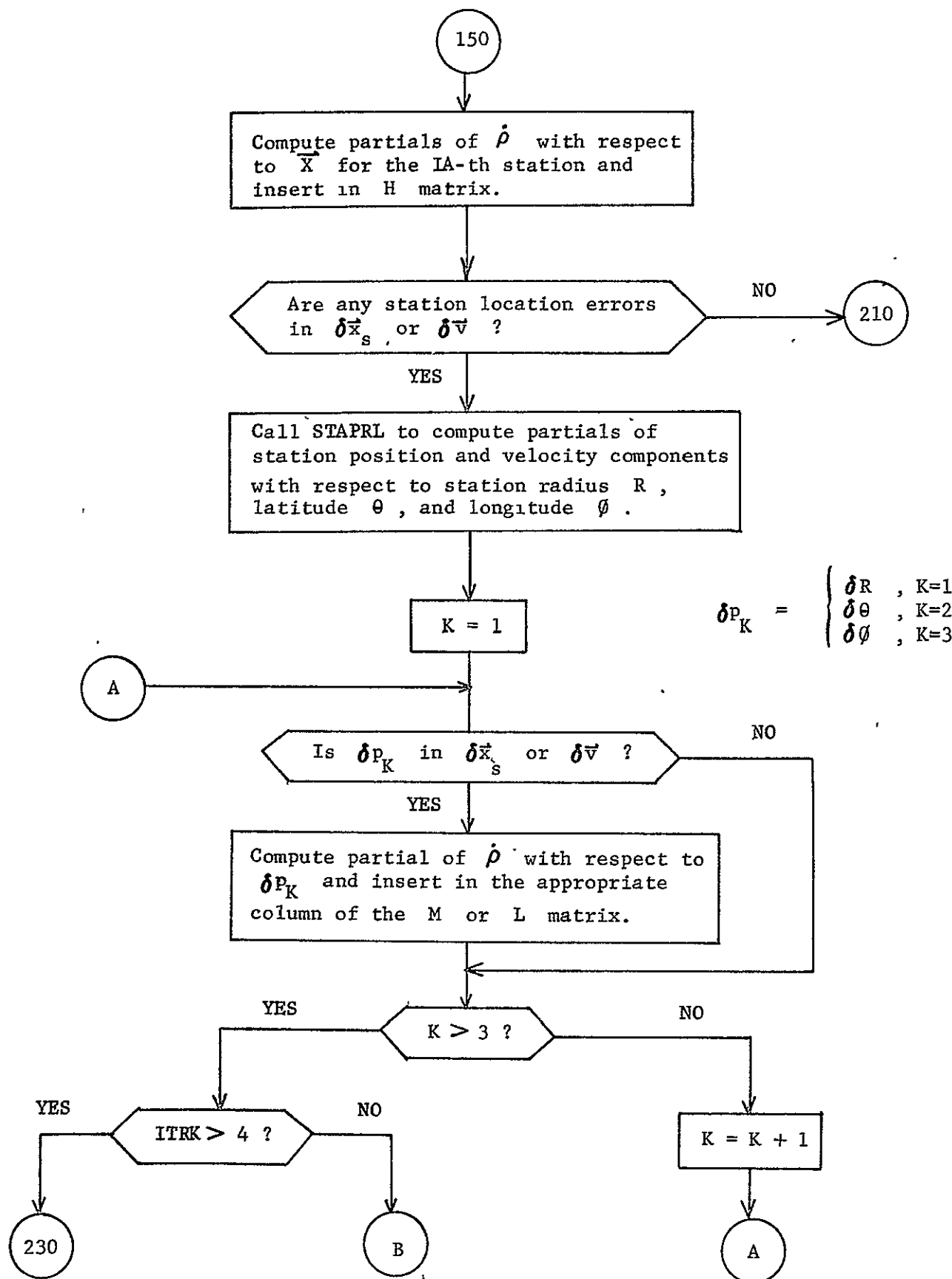
Partials of β with respect to \dot{X}_p , \dot{Y}_p , and \dot{Z}_p do not appear in the above expression since they are all zero. Partial of β with respect to the remaining target planet orbital elements are treated similarly.

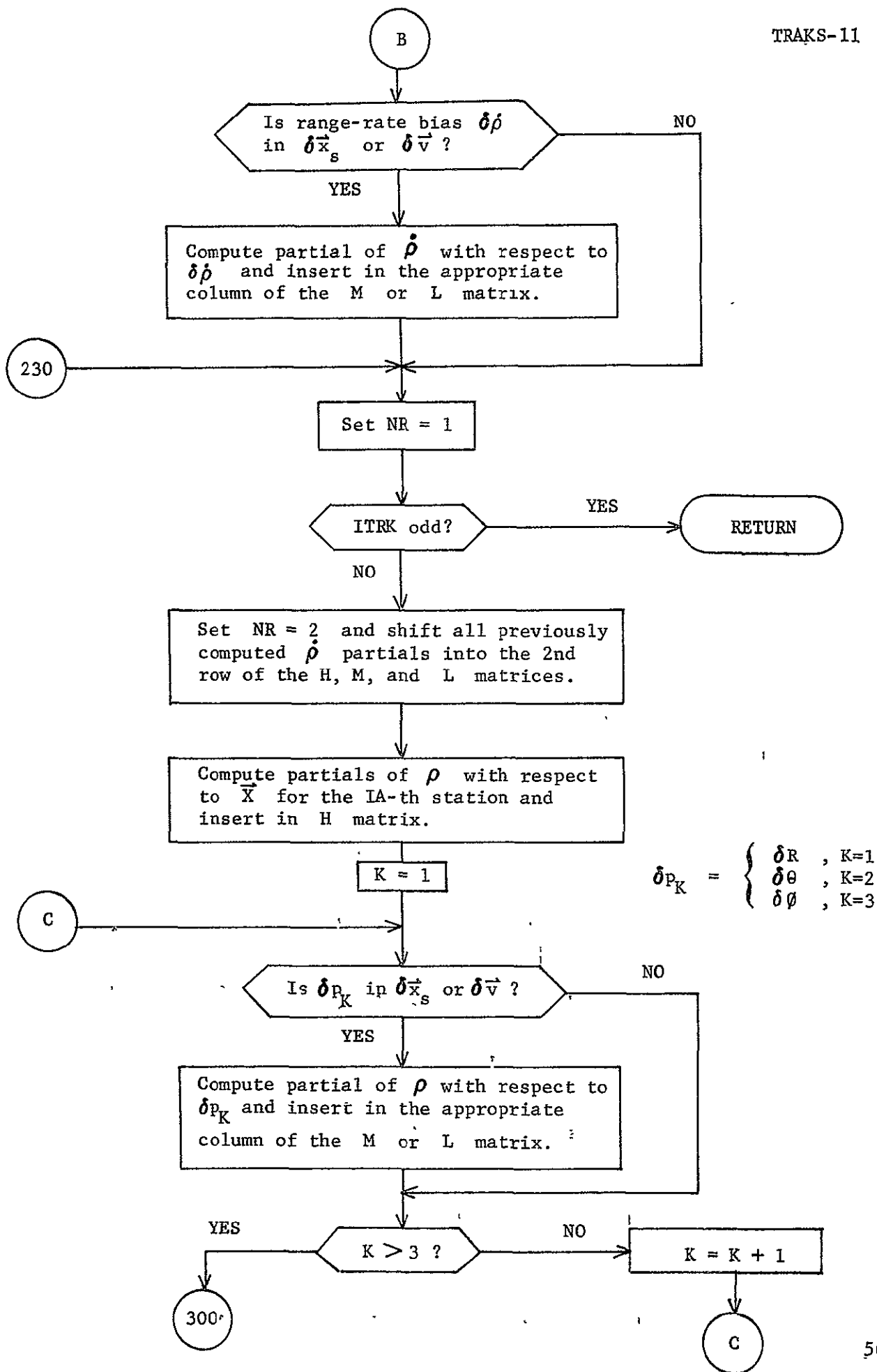
TRAKS Flow Chart

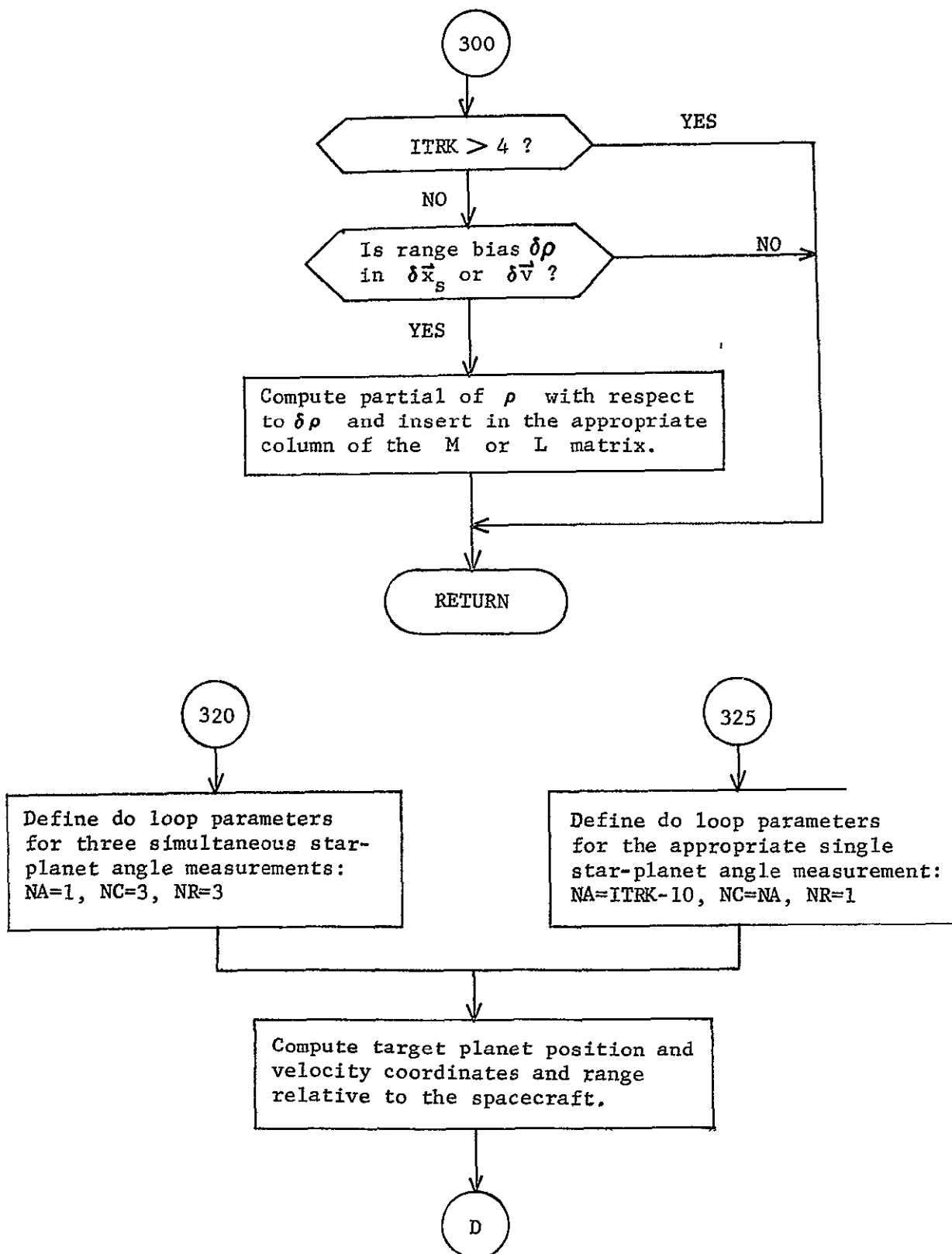


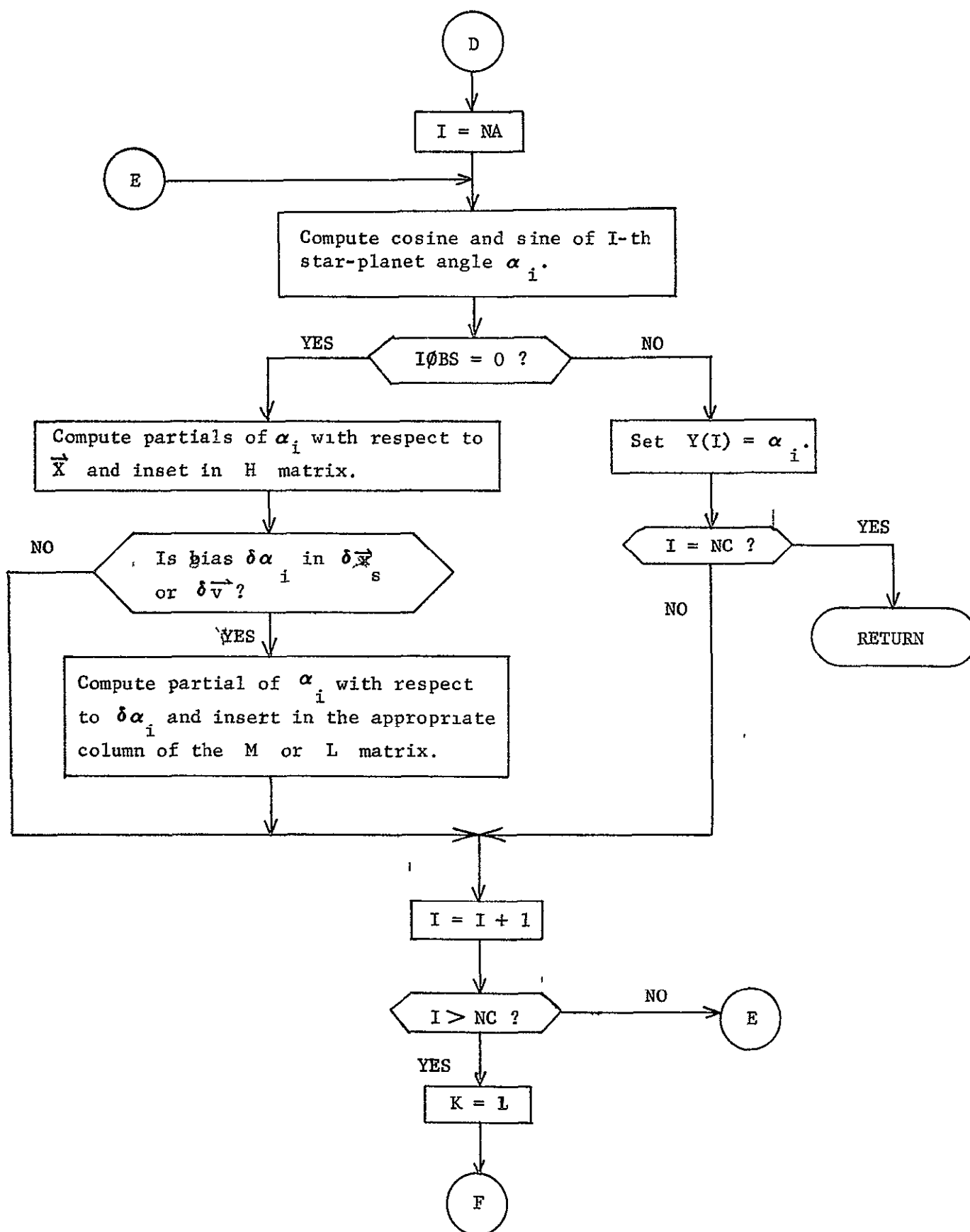




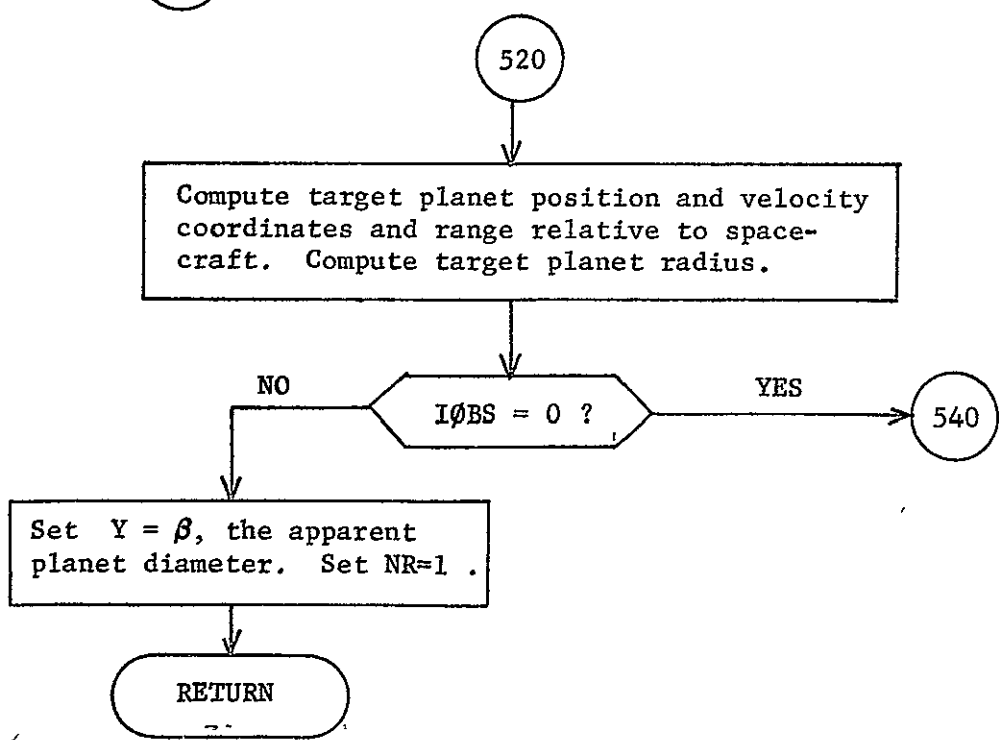
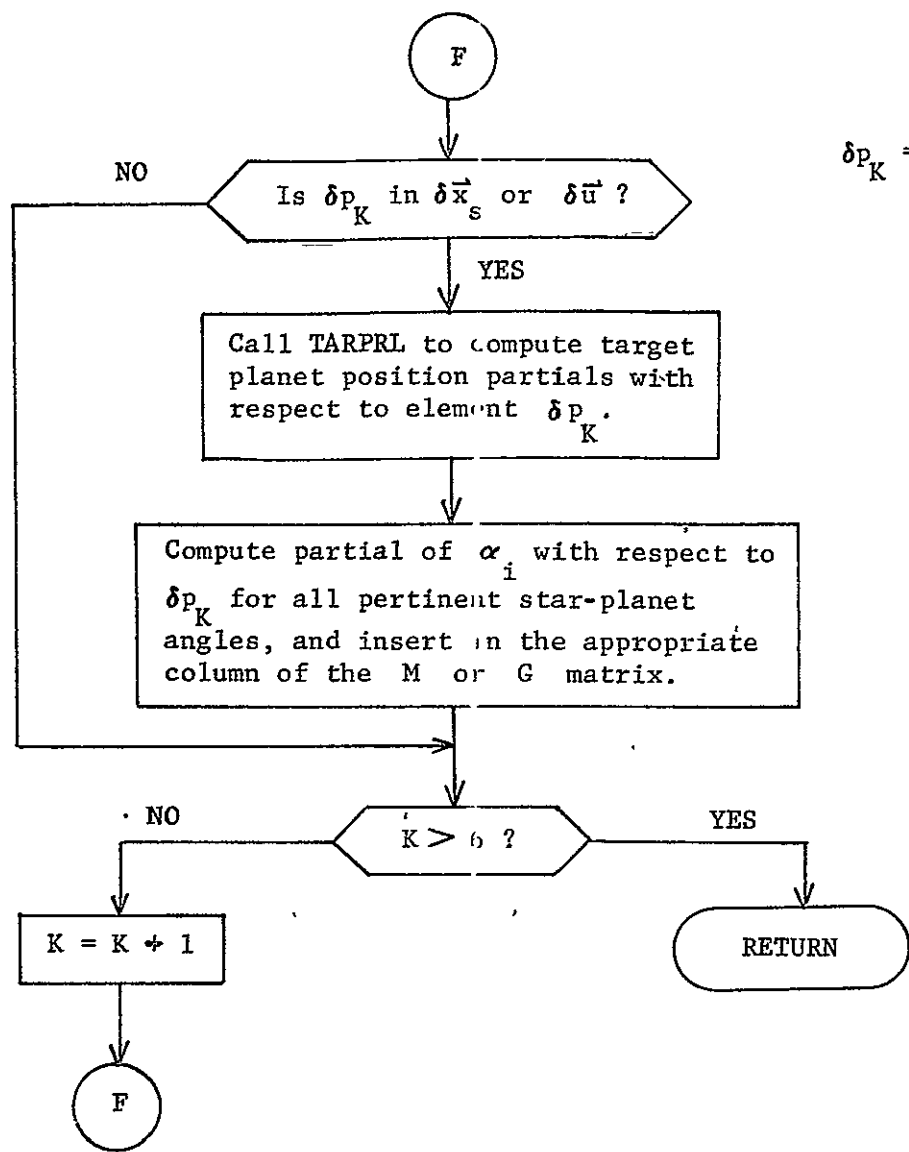


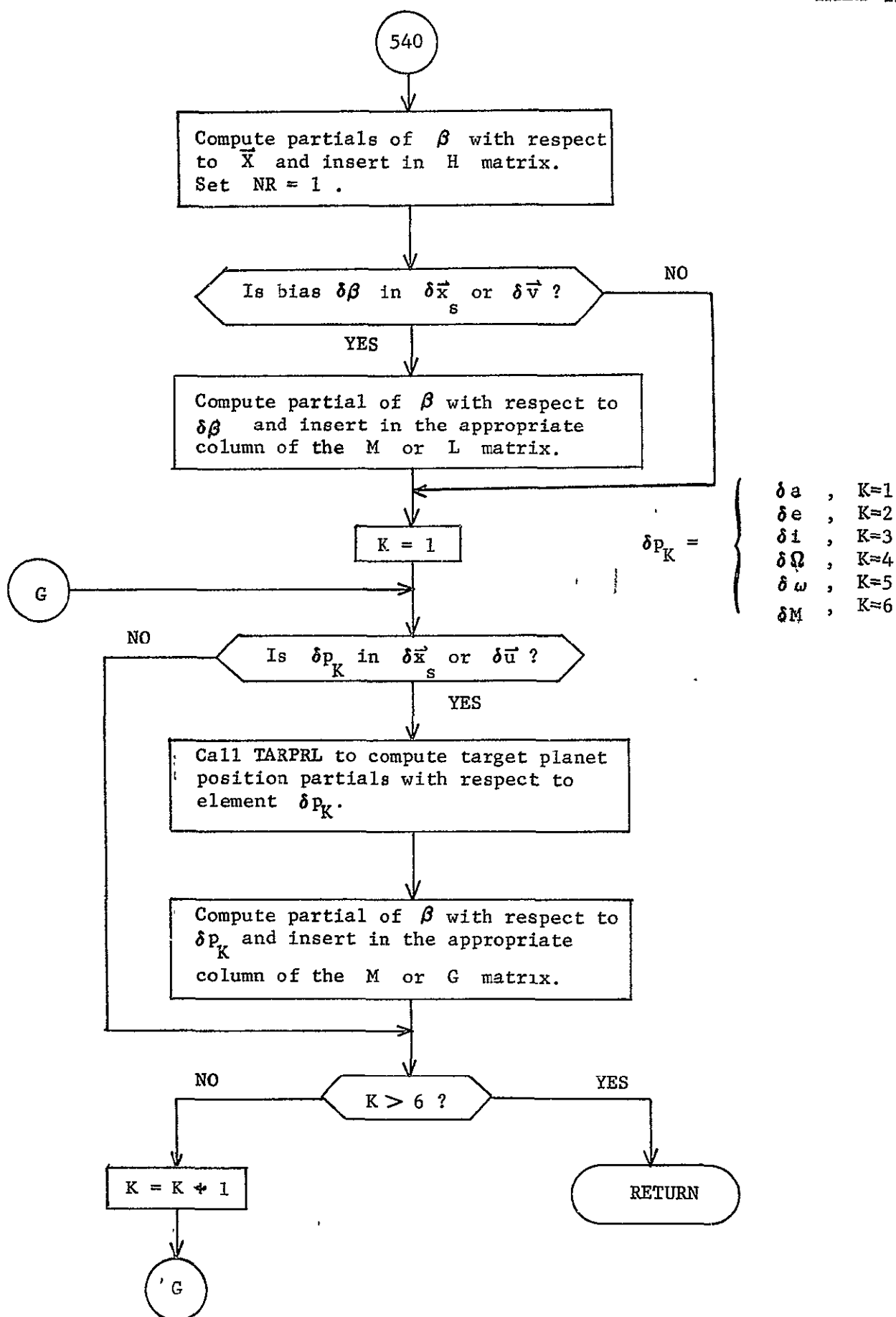






$$\delta p_K = \left\{ \begin{array}{ll} \delta a & , K=1 \\ \delta e & , K=2 \\ \delta i & , K=3 \\ \delta \Omega & , K=4 \\ \delta \omega & , K=5 \\ \delta M & , K=6 \end{array} \right.$$





SUBROUTINE TRANS

PURPOSE: TO PERFORM ONE OF THE FOLLOWING THREE OPTIONS.

1. CONVERT FROM GEOCENTRIC EQUATORIAL RECTANGULAR COORDINATES TO GEOCENTRIC ELIPTIC COORDINATES
2. CONVERT FROM GEOCENTRIC EQUATORIAL COORDINATES TO HELIOCENTRIC ECLIPTIC COORDINATES
3. CONVERT FROM GEOCENTRIC ECLIPTIC COORDINATES TO HELIOCENTRIC ECLIPTIC COORDINATES

CALLING SEQUENCE: CALL TRANS(ICODE,X,Y,Z,VX,VY,VZ,XE,YE,ZE,VXE,
VYE,VZE,EPS,ICODE2)

ARGUMENT: EPS I OBLIQUITY OF EARTH

ICODE I AN INTERNAL CODE THAT DETERMINES IF OPTION
1 OR 2 ABOVE WILL BE EXERCISED

ICODE2 I AN INTERNAL CODE THAT DETERMINES IF OPTION
3 ABOVE IS TO BE EXERCISED

VX I/O X-VELOCITY COMPONENT OF THE VEHICLE

VXE I X-VELOCITY COMPONENT OF EARTH IN
HELIOCENTRIC ECLIPTIC COORDINATES

VY I/O Y-VELOCITY COMPONENT OF THE VEHICLE

VYE I Y-VELOCITY COMPONENT OF EARTH

VZ I/O Z-VELOCITY COMPONENT OF THE VEHICLE

VZE I Z-VELOCITY COMPONENT OF EARTH

X I/O X-POSITION COMPONENT OF THE VEHICLE

XE I X-POSITION COMPONENT OF THE EARTH
IN HELIOCENTRIC ECLIPTIC COORDINATES

Y I/O Y-POSITION COMPONENT OF THE VEHICLE

YE I Y-POSITION COMPONENT OF THE EARTH

Z I/O Z-POSITION COMPONENT OF THE VEHICLE

ZE I Z-POSITION COMPONENT OF THE EARTH

SUBROUTINES SUPPORTED: DATA DATAS

LOCAL SYMBOLS: CE COSINE OF OBLIQUITY OF EARTH

DUM INTERMEDIATE VARIABLE

SE SINE OF OBLIQUITY OF EARTH

TRANS Analysis

Subroutine TRANS transforms the position and velocity components of the spacecraft from one coordinate system to another. The three options available with this subroutine are summarized below.

- 1) Convert from geocentric equatorial coordinates to geocentric ecliptic coordinates using the following equations:

$$\begin{aligned} X &= X \\ Y &= Y \cos \epsilon + Z \sin \epsilon \\ Z &= -Y \sin \epsilon + Z \cos \epsilon \\ \dot{X} &= \dot{X} \\ \dot{Y} &= \dot{Y} \cos \epsilon + \dot{Z} \sin \epsilon \\ \dot{Z} &= -\dot{Y} \sin \epsilon + \dot{Z} \cos \epsilon \end{aligned}$$

- 2) Convert from geocentric equatorial coordinates to heliocentric ecliptic coordinates. The same procedure as above is used to convert from geocentric equatorial to geocentric ecliptic. Then translate according to the following equations:

$$\begin{aligned} X &= X + X_E & \dot{X} &= \dot{X} + \dot{X}_E \\ Y &= Y + Y_E & \dot{Y} &= \dot{Y} + \dot{Y}_E \\ Z &= Z + Z_E & \dot{Z} &= \dot{Z} + \dot{Z}_E \end{aligned}$$

- 3) Convert from geocentric ecliptic coordinates to heliocentric ecliptic coordinates using the following equations:

$$\begin{aligned} X &= X + X_E & \dot{X} &= \dot{X} + \dot{X}_E \\ Y &= Y + Y_E & \dot{Y} &= \dot{Y} + \dot{Y}_E \\ Z &= Z + Z_E & \dot{Z} &= \dot{Z} + \dot{Z}_E \end{aligned}$$

SUBROUTINE TRAPAR

PURPOSE: TO COMPUTE THE FOLLOWING SET OF NAVIGATION PARAMETERS
 -- FLIGHT PATH ANGLE, ANGLE BETWEEN RELATIVE VELOCITY
 AND PLANE OF THE SKY, GEOCENTRIC DECLINATION, EARTH/
 SPACECRAFT/TARGET PLANET ANGLE, ANTENNA AXIS/LIMB OF
 SUN ANGLE, AND SPACECRAFT OCCULTATION RATIOS FOR SUN,
 MOON, AND PLANETS.

CALLING SEQUENCE: CALL TRAPAR

SUBROUTINES SUPPORTED: PRINT PRINT4 SETVEVS PRINT3 SETEVN

SUBROUTINES REQUIRED: EPHEM ORB PECEQ

LOCAL SYMBOLS: ALFA VECTOR FORMING RIGHT-HANDED ORTHOGONAL
 TRIAD WITH XN AND SSS VECTORS FOR
 CALCULATION OF ANTENNA AXIS/LIMB OF ANGLE
 OF SUN

AMAG MAGNITUDE OF THE ALFA VECTOR

BETA ANTENNA AXIS/EARTH ANGLE

CD COSINE OF GEOCENTRIC DECLINATION

CT INTERMEDIATE VARIABLE FOR ALL CALCULATIONS

CZAE COSINE OF EARTH/SPACECRAFT/TARGET PLANE
 ANGLE

DELTA GEOCENTRIC DECLINATION

DS INTERMEDIATE VARIABLE FOR CALCULATION OF
 OCCULTATION RATIOS

ECEQP TRANSFORMATION FROM EARTH ECLIPTIC TO
 EQUATORIAL FRAME FOR CALCULATION OF
 GEOCENTRIC DECLINATION

GAMMA INERTIAL FLIGHT PATH ANGLE

IND LOCATION IN THE F ARRAY OF THE EARTH
 POSITION AND VELOCITY IN THE INERTIAL
 FRAME

ISAVE SAVES AND RESTORES FIRST ELEMENT OF THE
 NO-ARRAY FOR BARYCENTRIC NAVIGATION

JND LOCATION IN THE F ARRAY OF THE TARGET
 PLANET POSITION AND VELOCITY IN THE
 INERTIAL FRAME

NINETY	CONSTANT VALUE, EQUAL TO 90.000
OCCULT	OCCULTATION RATIO OF THE I-TH PLANET
PHI	ANTENNA AXIS/LIMB OF SUN ANGLE
RDV	INTERMEDIATE VARIABLE, DOT PRODUCT OF TWO VECTORS
REMAG	MAGNITUDE OF THE EARTH HELIOCENTRIC POSITION
RIMAG	MAGNITUDE OF THE POSITION OF THE I-TH PLANET IN THE GEOCENTRIC ECLIPTIC FRAME
RMAG	MAGNITUDE OF THE SPACECRAFT HELIOCENTRIC POSITION
RSS	SPACECRAFT HELIOCENTRIC POSITION
SD	SINE OF GEOCENTRIC DECLINATION
SKYI	ANGLE BETWEEN SPACECRAFT VELOCITY RELATIVE TO EARTH AND PLANE OF THE SKY
SRDV	DOT PRODUCT OF SPACECRAFT GEOCENTRIC POSITION AND VELOCITY VECTORS
SRE	SPACECRAFT GEOCENTRIC ECLIPTIC POSITION AND VELOCITY
SRMAG	MAGNITUDE OF SPACECRAFT GEOCENTRIC POSITION
SRQ	SPACECRAFT GEOCENTRIC EQUATORIAL POSITION
SRTMAG	MAGNITUDE OF SRTIP VECTOR
SRTIP	SPACECRAFT ECLIPTIC POSITION RELATIVE TO TARGET PLANET
SVMAG	MAGNITUDE OF SPACECRAFT GEOCENTRIC VELOCITY
SX	INTERMEDIATE VARIABLE FOR ALL CALCULATIONS
SZAE	SINE OF EARTH/SPACECRAFT/TARGET PLANET ANGLE
THETA	INTERMEDIATE ANGLE USED TO CALCULATE NAVIGATION PARAMETERS
VMAG	MAGNITUDE OF SPACECRAFT VELOCITY RELATIVE

TO INERTIAL FRAME

XMAG MAGNITUDE OF THE XN VECTOR BEFORE
 UNITIZING

 XN CROSS PRODUCT OF SPACECRAFT GEOCENTRIC
 POSITION AND SPACECRAFT SPIN AXIS

 ZAE EARTH/SPACECRAFT/TARGET PLANET ANGLE

COMMON COMPUTED/USED:

B NO RE

COMMON USED:

F	IBARY	NBOD	NB	NTP
ONE	PLANET	RADIUS	RAD	SSS
TWO	V	XP	ZERO	

TRAPAR Analysis

The coordinate systems and variables required for the derivation of the first four navigation parameters are shown in Figure 1. The inertial coordinate system XYZ may be heliocentric or barycentric ecliptic. The position and velocity of the earth in inertial space is given by \vec{r}_E and \vec{v}_E ; that of the spacecraft, by \vec{r} and \vec{v} ; and that of the target planet (or moon), by \vec{r}_{TP} and \vec{v}_{TP} . The xyz coordinate system is geocentric equatorial.

1. Flight path angle, γ .

Let θ denote the angle between \vec{r} and \vec{v} , so that

$$\cos \theta = \frac{\vec{r} \cdot \vec{v}}{r v} \quad \text{and} \quad \sin \theta = + \left[1 - \cos^2 \theta \right]^{\frac{1}{2}}.$$

Then

$$\gamma = \frac{\pi}{2} - \theta.$$

2. Angle between relative velocity and plane of the sky, i' .

The plane of the sky is defined as the plane perpendicular to the vector $\vec{r} - \vec{r}_E$. Let θ' denote the angle between $\vec{r} - \vec{r}_E$ and $\vec{v} - \vec{v}_E$, so that

$$\cos \theta' = \frac{(\vec{r} - \vec{r}_E) \cdot (\vec{v} - \vec{v}_E)}{|\vec{r} - \vec{r}_E| |\vec{v} - \vec{v}_E|} \quad \text{and} \quad \sin \theta' = + \left[1 - \cos^2 \theta' \right]^{\frac{1}{2}}$$

Then

$$i' = \frac{\pi}{2} - \theta'.$$

Note that, i' is not defined if the relative velocity $\vec{v} - \vec{v}_E$ is zero.

3. Geocentric declination, δ .

Let (x, y, z) denote the geocentric equatorial components of $\vec{r} - \vec{r}_E$. Then

$$\delta = \tan^{-1} \left(\frac{z}{\sqrt{x^2 + y^2}} \right).$$

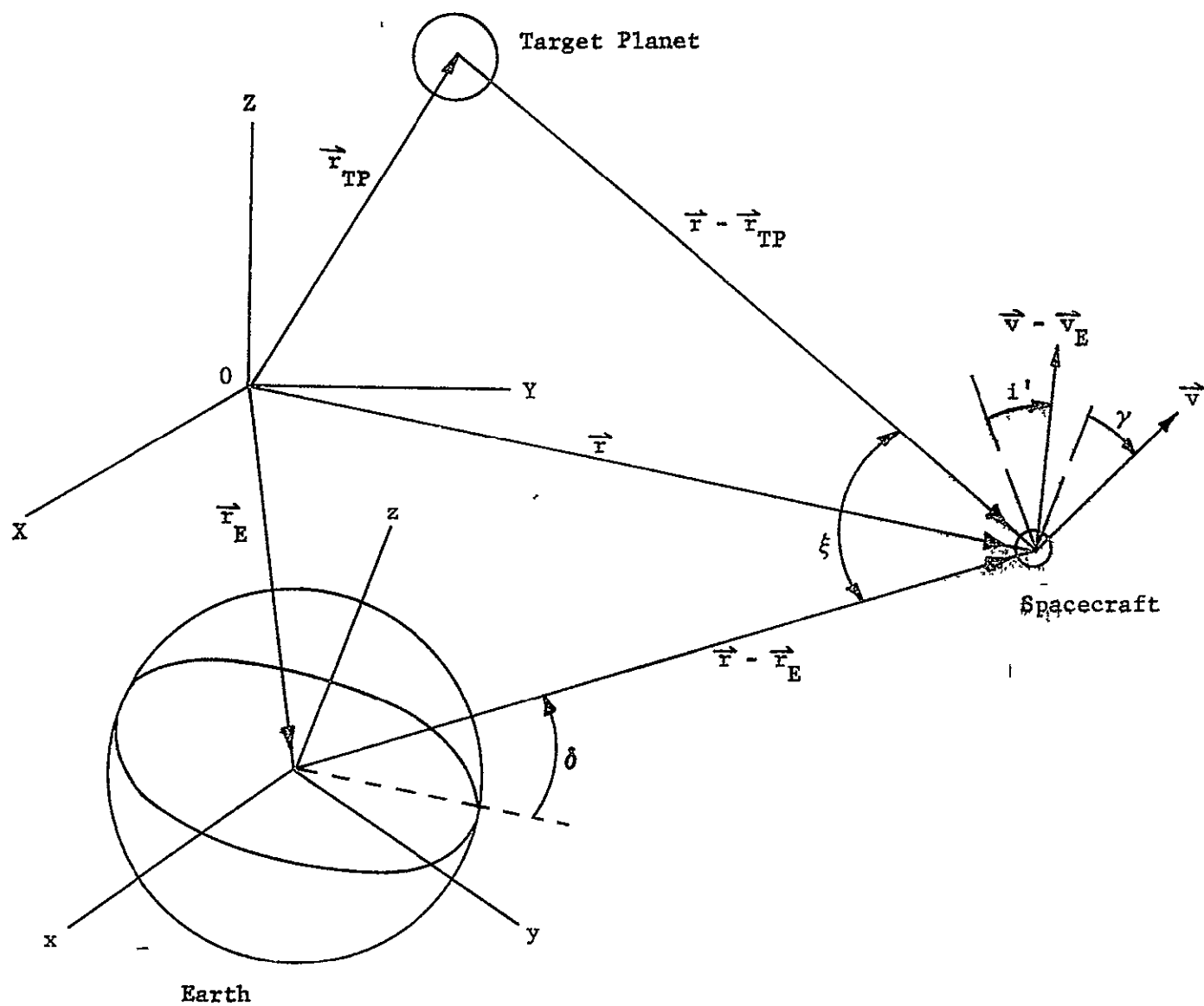


Figure 1

4. Earth/spacecraft/target planet angle, ξ .

The angle ξ is the angle between the vectors $\vec{r} - \vec{r}_E$ and $\vec{r} - \vec{r}_{TP}$, so that

$$\cos \xi = \frac{(\vec{r} - \vec{r}_E) \cdot (\vec{r} - \vec{r}_{TP})}{|\vec{r} - \vec{r}_E| |\vec{r} - \vec{r}_{TP}|}$$

$$\text{and} \quad \sin \xi = + \left[1 - \cos^2 \xi \right]^{\frac{1}{2}}.$$

The next two navigation parameters relate to the spacecraft antenna axis. The pertinent geometry is shown in Figure 2. The antenna axis $\vec{\alpha}$ is defined as the intersection between the antenna plane (the plane perpendicular to the spacecraft spin axis \vec{s}) and the plane formed by the $\vec{r} - \vec{r}_E$ and \vec{s} vectors. The vector $\vec{\rho}$ originates from the limb of the sun and lies in the $\vec{r}, \vec{\alpha}$ plane.

5. Antenna axis/Earth angle, ψ .

Let ψ denote the angle between the unit spin axis vector \vec{s} and $\vec{r} - \vec{r}_E$, so that

$$\cos \psi = \frac{\vec{s} \cdot (\vec{r} - \vec{r}_E)}{|\vec{r} - \vec{r}_E|} \quad \text{and} \quad \sin \psi = + \left[1 - \cos^2 \psi \right]^{\frac{1}{2}}.$$

Then

$$\beta = \frac{\pi}{2} - \psi.$$

Note that the antenna axis is not uniquely defined when the angle $\psi = 0$.

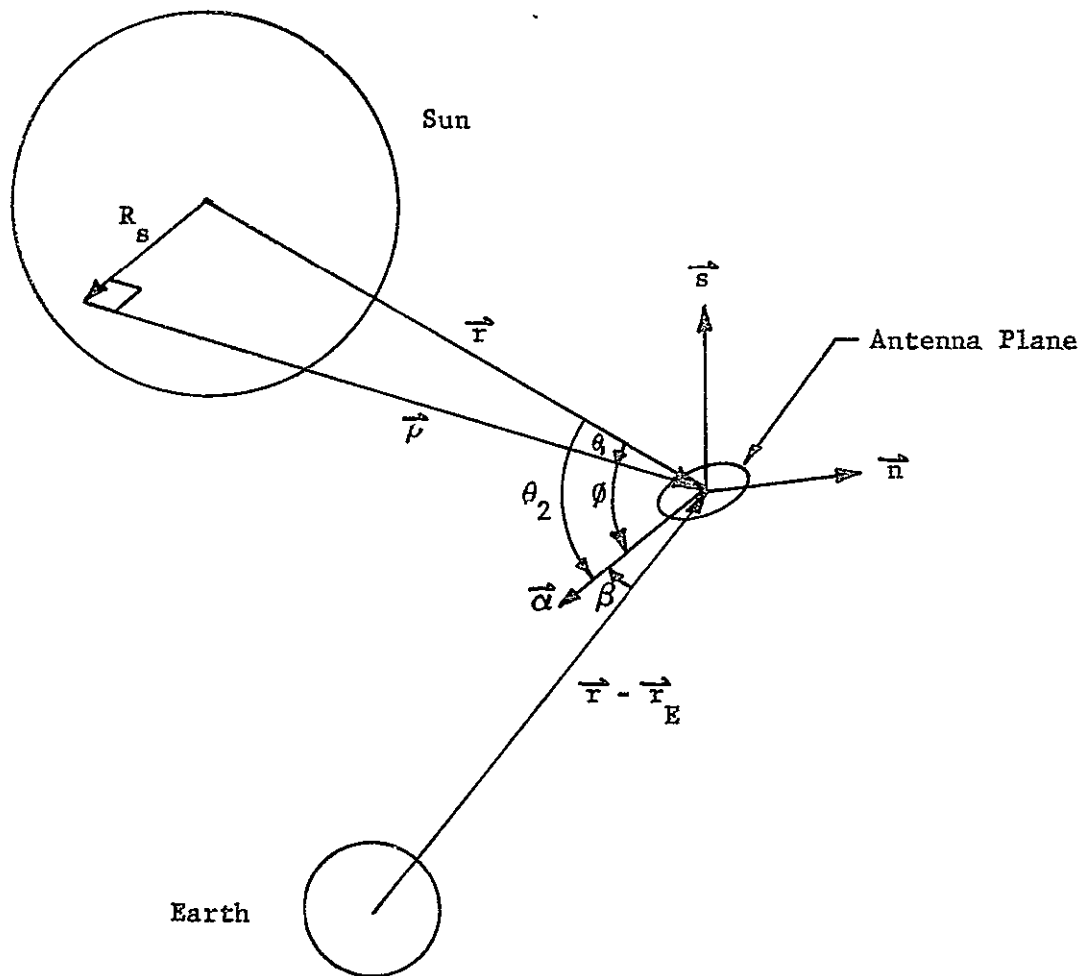


Figure 2. Antenna Axis Geometry

6. Antenna axis/limb of Sun angle, ϕ .

The unit vector \vec{n} normal to the \vec{s} , $\vec{r} - \vec{r}_E$ plane is given by

$$\vec{n} = \frac{(\vec{r} - \vec{r}_E) \times \vec{s}}{|(\vec{r} - \vec{r}_E) \times \vec{s}|}.$$

Then the unit antenna axis vector $\vec{\alpha}$ is given by

$$\vec{\alpha} = \vec{n} \times \vec{s}.$$

The angle θ_2 denotes the angle between the vectors \vec{r} and $\vec{\alpha}$, so that

$$\cos \theta_2 = - \frac{\vec{r} \cdot \vec{\alpha}}{r} \quad \text{and} \quad \sin \theta_2 = + \left[1 - \cos^2 \theta_2 \right]^{\frac{1}{2}}.$$

The angle θ_1 denotes the angle between the vectors $\vec{\rho}$ and \vec{r} , so that

$$\theta_1 = \sin^{-1} \left(\frac{R_s}{r} \right), \quad 0 \leq \theta_1 \leq \frac{\pi}{2}$$

where R_s is the radius of the Sun.

Then

$$\phi = \theta_2 - \theta_1.$$

The final set of navigation parameters relate to spacecraft occultation ratios for the Sun and all other celestial bodies assumed in the dynamic model. The pertinent geometry is shown in Figure 3. The position of the i -th celestial body relative to the Sun is denoted by \vec{r}_i . Occultation parameters d_s and d_i are defined as the minimal distances from the centers of the Sun and i -th body, respectively, to the Earth/spacecraft vector $\vec{r} - \vec{r}_E$.

7. Spacecraft occultation ratio for the Sun.

The occultation ratio for the Sun is defined as d_s/R_s , where R_s is the Sun radius. As long as the occultation ratio is greater than one, the spacecraft is neither being occulted by the Sun nor passing in front of the Sun. The occultation ratio is computed only when the angle between the $\vec{r} - \vec{r}_E$ and \vec{r}_E vectors is less than or equal to 90 degrees, or, equivalently, when

$$\vec{r}_E \cdot (\vec{r} - \vec{r}_E) \geq 0.$$

If this condition is satisfied, the occultation ratio is computed using the equations

$$d_s = \left[r_E^2 - b^2 \right]^{\frac{1}{2}}$$

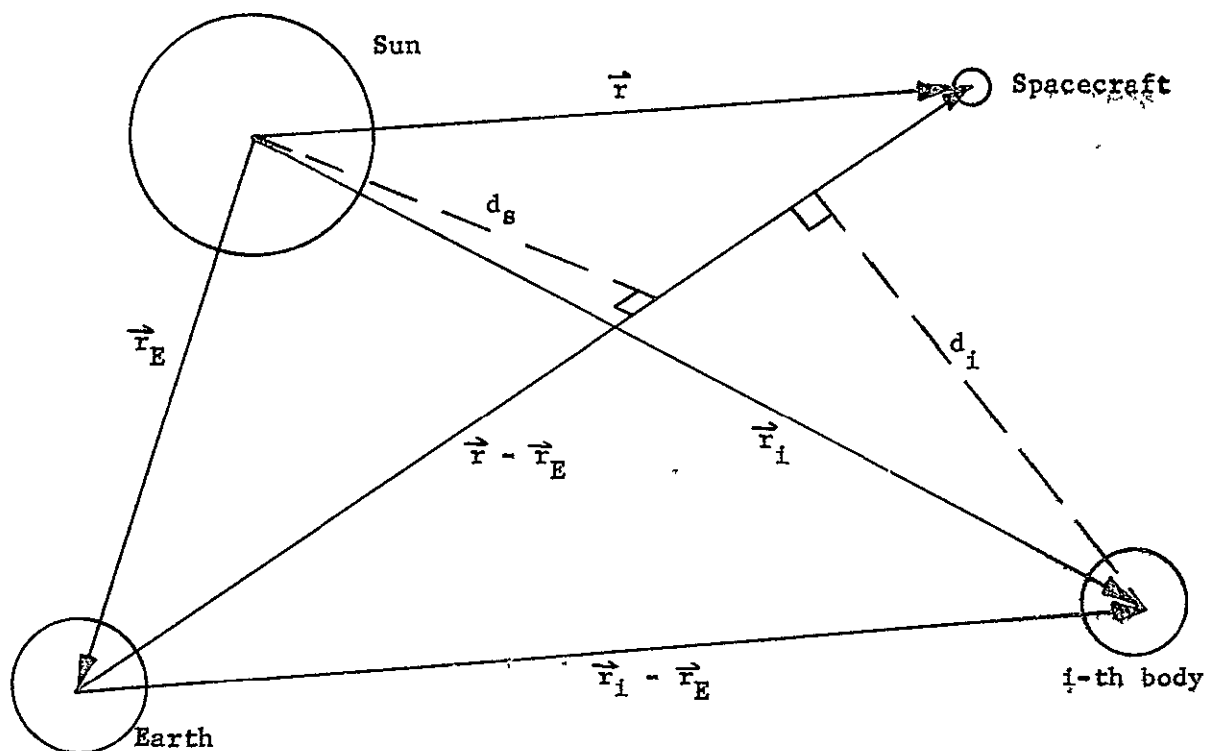


Figure 3. Occultation Geometry

and

$$b = \frac{-\vec{r}_E \cdot (\vec{r} - \vec{r}_E)}{|\vec{r} - \vec{r}_E|}.$$

Occultation occurs if $\frac{d_s}{R_s} \leq 1$ and $|\vec{r} - \vec{r}_E| \geq r_E$; if $\frac{d_s}{R_s} \leq 1$ and $|\vec{r} - \vec{r}_E| < r_E$, then the spacecraft is passing in front of the Sun.

8. Spacecraft occultation ratios for other celestial bodies.

The occultation ratio for the i-th celestial body is defined as d_i/R_i , where R_i is the radius of the i-th body. The occultation ratio is

computed only when

$$(\vec{r} - \vec{r}_E) \cdot (\vec{r}_1 - \vec{r}_E) \geq 0 \quad .$$

If this conditions is satisfied, the occultation ratio is computed using the equations

$$d_i = \left[a_i^2 - b_i^2 \right]^{\frac{1}{2}}$$

$$a_i = \left| \vec{r}_1 - \vec{r}_E \right|$$

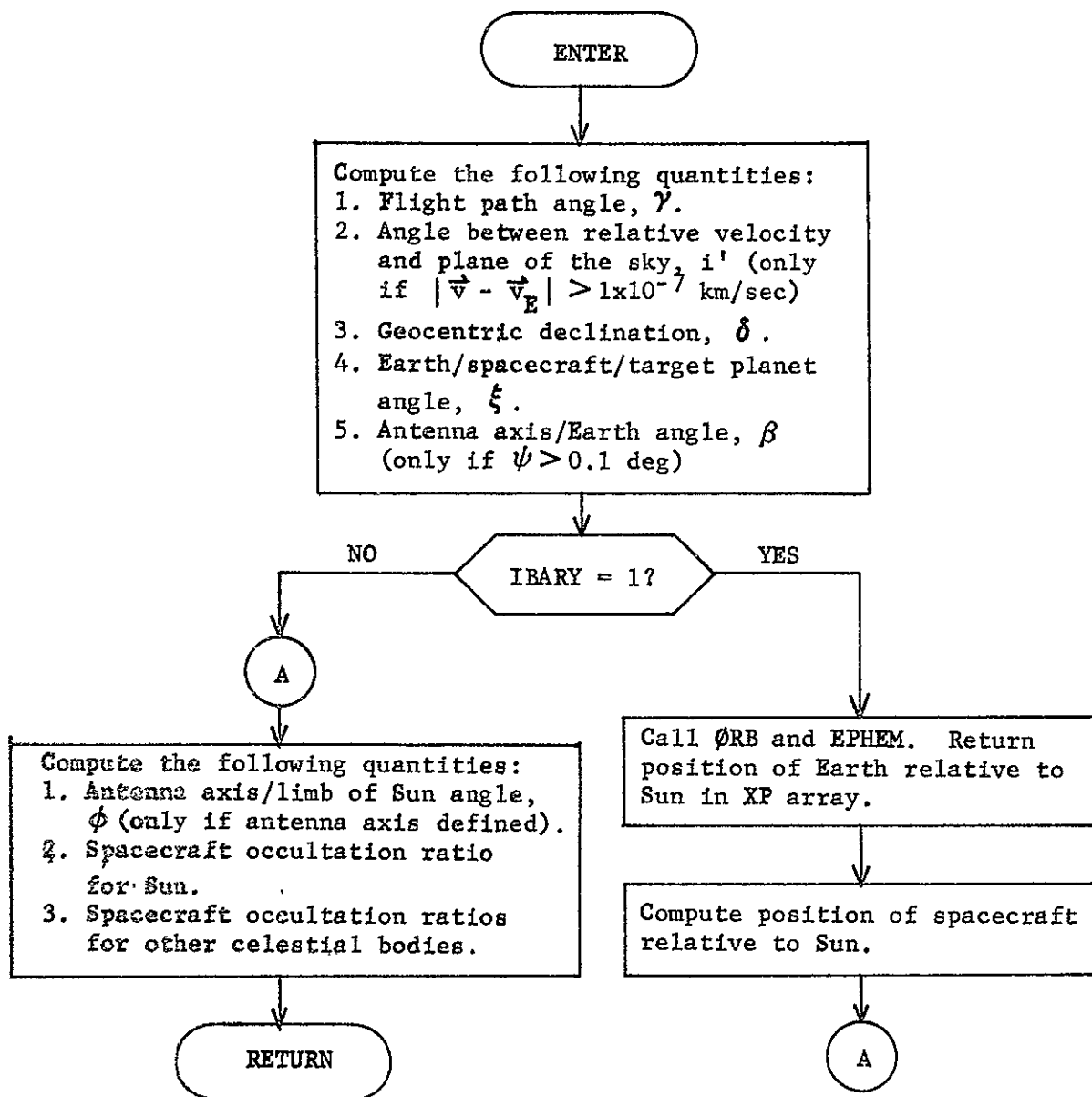
and

$$b_i = \frac{(\vec{r} - \vec{r}_E) \cdot (\vec{r}_1 - \vec{r}_E)}{\left| \vec{r} - \vec{r}_E \right|} \quad .$$

Occultation occurs if $\frac{d_i}{R_i} \leq 1$ and $\left| \vec{r} - \vec{r}_E \right| \geq \left| \vec{r}_1 - \vec{r}_E \right|$; if

$\frac{d_i}{R_i} \leq 1$ and $\left| \vec{r} - \vec{r}_E \right| < \left| \vec{r}_1 - \vec{r}_E \right|$, then the spacecraft is passing is front of the i-th celestial body.

TRAPAR Flow Chart



SUBROUTINE TRJTRY

PURPOSE: TO DETERMINE THE TIME OF THE NEXT GUIDANCE EVENT AND
INTEGRATE THE NOMINAL TRAJECTORY FROM THE PREVIOUS
EVENT TIME TO THE NEXT TIME.

CALLING SEQUENCE: CALL TRJTRY

SUBROUTINES SUPPORTED: NOMNAL

SUBROUTINES REQUIRED: VMP

LOCAL SYMBOLS: ACK ACCURACY USED TO INTEGRATE THE NOMINAL
TRAJECTORY

DELMIN TIME(DAYS) BETWEEN THE LAST EVENT AND THE
NEXT EVENT

DELTM SAME AS DELMIN -THE TIME VMP IS TO
INTEGRATE THE TRAJECTORY UNLESS ANOTHER
STOPPING CONDITION OCCURS

DELT SAME AS DELMIN

ERROR MINIMUM ALLOWABLE VALUE OF DELMIN

ISP2 FLAG TO CONTROL STOPPING CONDITION
=1 STOP AT SHPERE-OF-INFLUENCE
=0 DO NOT STOP AT SHPERE-OF-INFLUENCE

I INDEX

J INDEX

RSF SPACECRAFT STATE AT FINAL TIME

COMMON COMPUTED/USED:	D1	ICL2	ICL	ISPH	KSICA
	KTIM	RIN	TIMG	TRTM	

COMMON COMPUTED:	DELTP	IEPHM	INPR	IPRINT	KUR
------------------	-------	-------	------	--------	-----

COMMON USED:	ACKT	KGYD	NCPR	NOGYD	TMPR
	V				

TRJTRY Analysis

TRJTRY determines the time of the next guidance event and integrates the nominal trajectory from the previous event time to the next time.

Special provisions must be made in determining the next guidance event because of the flexibility permitted in specifying the times of those guidance events. For every guidance event i , parameters $KTIM(i)$ and $TIMG(i)$ will have been set before entering TRJTRY. $KTIM(i)$ prescribes the epoch to which the guidance event i is referenced with $KTIM(i) = 1, 2, 3$ corresponding to epochs of initial time, sphere of influence (SOI) intersection, and closest approach (CA) passage respectively. $TIMG(i)$ then specifies the time interval (days) from the epoch to the guidance event. The guidance events do not need to be arranged chronologically. After execution of each guidance event i the flag $KTIM(i)$ is set equal to 0.

The first computational procedure in TRJTRY is the sequencing loop. Here a search determines the minimum value of $TIMG(i)$ over all values of i such that $KTIM(i) = 1$. The time interval Δt between that time and the current time is then computed. If Δt is less than an allowable tolerance ϵ ($=10^{-5}$ days) the program returns to NOMNAL for the processing of the current event.

If $\Delta t \geq \epsilon$ TRJTRY must perform an integration to the next guidance event. TRJTRY first sets up flags controlling integration stopping conditions depending upon the current value of $KSICA$. The flag $KSICA$ determines the current phase of the trajectory. $KSICA$ is initially set equal to 1 (PRELIM). When the target planet SOI is encountered $KSICA$ is set to 2. Finally when CA to the target planet occurs it is set to 3.

The stopping condition flags are $ISP2$ and $ICL2$. The flag $ISP2$ determines whether the integration should be stopped at SOI if encountered ($ISP2 = 1$) or not ($ISP2 = 0$). The flag $ICL2$ determines whether the integration should be stopped at CA if encountered ($ICL2 = 1$) or not ($ICL2 = 0$).

Therefore if $KSICA = 1$, TRJTRY sets $ISP2 = 1$ so that the integration will stop at the guidance event time only if that time occurs before SOI. But if the SOI is encountered before the event time, all times referenced to the SOI must be updated before determining the next event. Similarly when $KSICA = 2$ TRJTRY sets $ICL2 = 1$ so that times referenced to CA may be updated when CA occurs. Of course when $KSICA = 3$, all times have been updated (referenced to initial time) and neither $ISP2$ nor $ICL2$ need be set to 1.

Having set the stopping condition flags, TRJTRY now calls VMP for the propagation of the trajectory to the required stopping condition. At the end of the integration it records the current trajectory time and state.

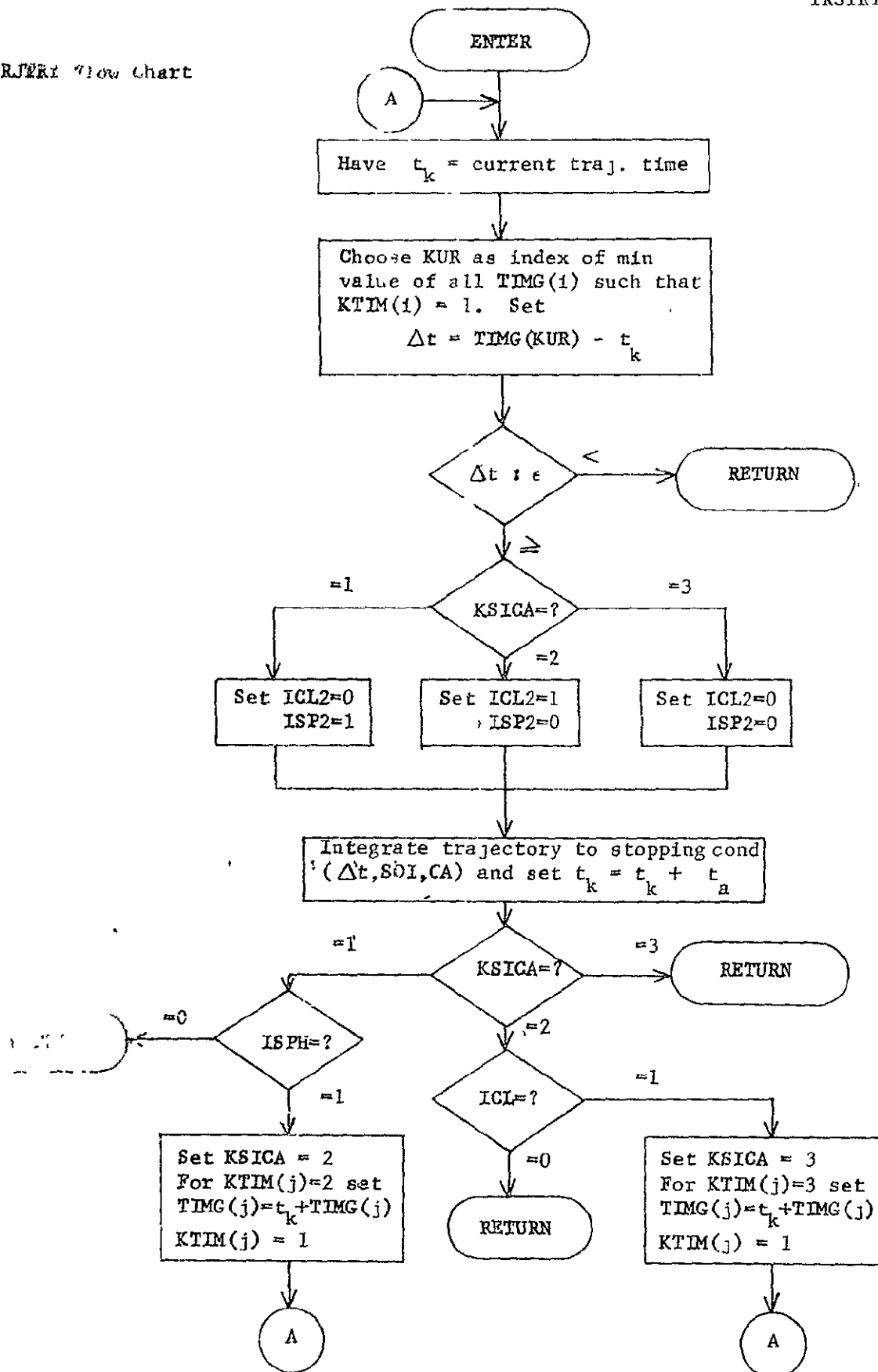
TRJTRY now sorts again on $KSICA$. If $KSICA = 3$, the trajectory has been integrated to the time of the current event and so control may be returned to NOMNAL.

If KSICA = 1 the SOI had not yet been reached at the previous event. TRJTRY then checks the flag ISPH. The flag ISPH reveals whether the current trajectory intersected the target planet SOI (ISPH = 1) or did not (ISPH = 0). Therefore if ISPH = 0, the current guidance event occurred before the trajectory intersected the SOI and thus the current state corresponds to the time of the guidance event. Therefore the return is made to NOMNAL.

If however KSICA = 1 and ISPH = 1 the trajectory integration was stopped at the SOI. TRJTRY now sets KSICA = 2 and updates all times referenced to the SOI so that they are now referenced to initial time (KTIM(i) = 1). It reenters the sequencing loop to determine the time of the next guidance event where the candidate events now include those originally referenced to SOI.

Similar steps are made when KSICA = 2. The flag ICL designates whether the current trajectory had a CA (ICL = 1) or not (ICL = 0). If KSICA = 2 and ICL = 0, the trajectory encountered the guidance event before reaching a CA so the return is made to NOMNAL. If KSICA = 2 and ICL = 1, the final time and state of the trajectory refer to closest approach. In this case TRJTRY sets KSICA = 3 and updates to initial time all times originally referenced to CA. It then returns to the sequencing loop.

TRJTRY Flow Chart



SUBROUTINE USCALE

PURPOSE: SCALES LENGTH OF A THREE-VECTOR TO SPECIFIED NUMBER

ARGUMENT: A I LENGTH OF VECTOR TO BE OUTPUT

U I VECTOR TO BE SCALED

V O SCALED OUTPUT VECTOR

SUBROUTINES SUPPORTED: CAREL IMPCT KTROL STIMP SUBSOL

SUBROUTINE UXV

PURPOSE: CALCULATE VECTOR CROSS PRODUCT OF TWO VECTORS

ARGUMENT: UV O CROSS PRODUCT OF U AND V

U I INPUT VECTOR

V I INPUT VECTOR

SUBROUTINES SUPPORTED: CAREL IMPCT KTROL STIMP SUBSOL

SUBROUTINE VARADA

PURPOSE COMPUTE VARIATION MATRIX FOR THREE-VARIABLE 8-PLANE
GUIDANCE POLICY IN THE ERROR ANALYSIS PROGRAM

CALLING SEQUENCE: CALL VARADA(RI, XSIP, XSIV, TEVN, TSI, ADA, BS, BDRS,
BDRS)

ARGUMENT: ADA O VARIATION MATRIX

BS I B OF THE NOMINAL TRAJECTORY

BDRS I B DOT R OF THE NOMINAL TRAJECTORY

BDRS I B DOT T OF THE NOMINAL TRAJECTORY

RI I POSITION AND VELOCITY OF THE VEHICLE AT THE
TIME OF GUIDANCE EVENT

TEVN I TRAJECTORY TIME OF THE GUIDANCE EVENT

TSI I TRAJECTORY TIME AT WHICH THE VEHICLE
REACHED THE SPHERE OF INFLUENCE ON THE
NOMINAL TRAJECTORY

XSIP I POSITION OF THE VEHICLE AT THE SPHERE OF
INFLUENCE ON THE NOMINAL TRAJECTORY

XSIV I VELOCITY OF THE VEHICLE AT THE SPHERE OF
INFLUENCE ON THE NOMINAL TRAJECTORY

SUBROUTINES SUPPORTED: GUID

SUBROUTINES REQUIRED: NTM

LOCAL SYMBOLS: BDR1 TEMPORARY STORAGE FOR BDR

BDT1 TEMPORARY STORAGE FOR BDT

B1 TEMPORARY STORAGE FOR B

DSI1 TEMPORARY STORAGE FOR DSI

IPR TEMPORARY STORAGE FOR IPRINT

ISP TEMPORARY STORAGE FOR ISP2

IPO TEMPORARY STORAGE FOR IPRINT

RF ALTERED FINAL STATE OF VEHICLE

TSI1 TEMPORARY STORAGE FOR TSI

XC ALTERED INITIAL STATE OF VEHICLE

COMMON COMPUTED/USED:	BDR ISP2	BDT	DSI	IPRINT	ISPH
COMMON COMPUTED:	B	DELTH	RSI	TRTH1	VSI
COMMON USED:	DATEJ NTP	FACP	FACV	FNTM	NTMC

VARADA Analysis

Subroutine VARADA employs numerical differencing to compute the variation matrix η for the three-variable B-plane guidance policy in the guidance event of the error analysis mode. See subroutine VARSIM Analysis for further analytical details, since the only difference between VARADA and VARSIM is that VARADA computations are based on the most recent targeted nominal, while VARSIM computations are based on the most recent nominal. The VARADA flow chart is identical to that of VARSIM except for the fact that in VARADA the nominal position/velocity state at t_{SI} is saved prior to calling VARADA, while in VARSIM it is saved locally.

SUBROUTINE VARSIM

**PURPOSE COMPUTE VARIATION MATRIX FOR THREE-VARIABLE B-PLANE
GUIDANCE POLICY IN THE SIMULATION PROGRAM**

CALLING SEQUENCE: CALL VARSIM(RI1,TEVN,TSI,ADA)

ARGUMENT: ADA 0 VARIATION MATRIX

**RI1 I VEHICLE POSITION/VELOCITY ON MOST RECENT
 NOMINAL TRAJECTORY AT TIME OF THE GUIDANCE
 EVENT**

TEVN I TRAJECTORY TIME OF GUIDANCE EVENT

TSI I TRAJECTORY TIME AT SPHERE OF INFLUENCE

SUBROUTINES SUPPORTED: GUISS

SUBROUTINES REQUIRED: NTMS

LOCAL SYMBOLS: BDRS TEMPORARY STORAGE FOR BDR

BDTs TEMPORARY STORAGE FOR BDT

BS TEMPORARY STORAGE FOR B

IPR TEMPORARY STORAGE FOR IPRINT

ISPS TEMPORARY STORAGE FOR ISP2

**RF1 ALTERED FINAL STATE OF VEHICLE ON MOST
 RECENT NOMINAL**

RSIS TEMPORARY STORAGE FOR RSI

TSI1 TEMPORARY STORAGE FOR TSI

VSIS TEMPORARY STORAGE FOR VSI

**XC ALTERED INITIAL STATE OF VEHICLE ON MOST
 RECENT NOMINAL**

COMMON COMPUTED/USED: BDR BDT B DSI IPRINT

ISPH ISP2 RSI VSI

COMMON COMPUTED: TRTM1

COMMON USED: DATEJ FACP FACV NTMC NTP

VARSIM Analysis

Subroutine VARSIM employs numerical differencing to compute the variation matrix η for the three-variable B-plane guidance policy in the guidance event of the simulation mode. This variation matrix relates deviations in the position/velocity state at t_k to deviations in $B \cdot T$, $B \cdot R$, and t_{SI} :

$$\begin{bmatrix} \delta B \cdot T \\ \delta B \cdot R \\ \delta t_{SI} \end{bmatrix} = \eta \begin{bmatrix} \delta \vec{R}_k \\ \delta \vec{V}_k \end{bmatrix} = \eta \delta \vec{X}_k$$

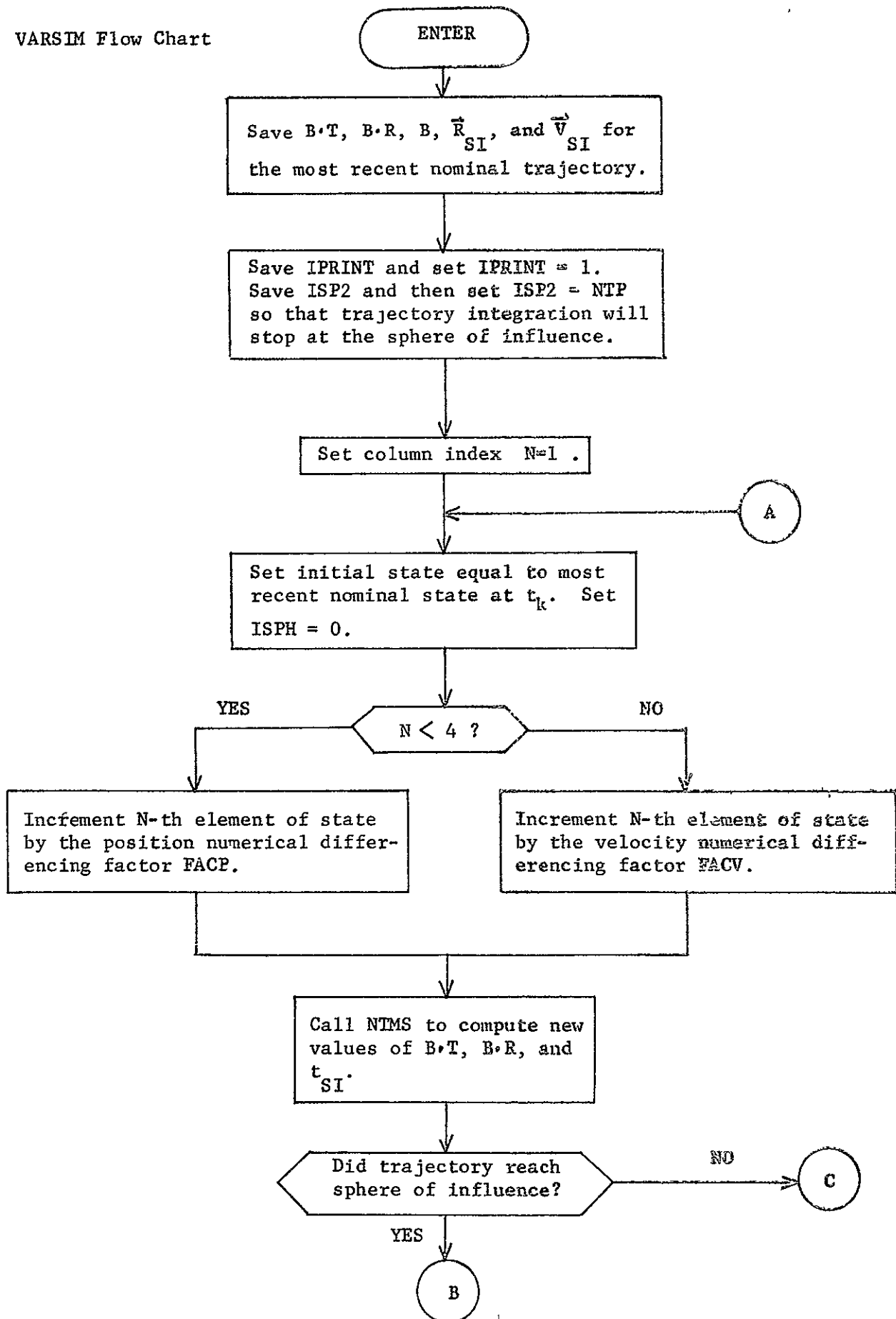
Since no good analytical formulas which relate δt_{SI} to $\delta \vec{R}_k$ and $\delta \vec{V}_k$ exist, numerical differencing must be employed to compute η .

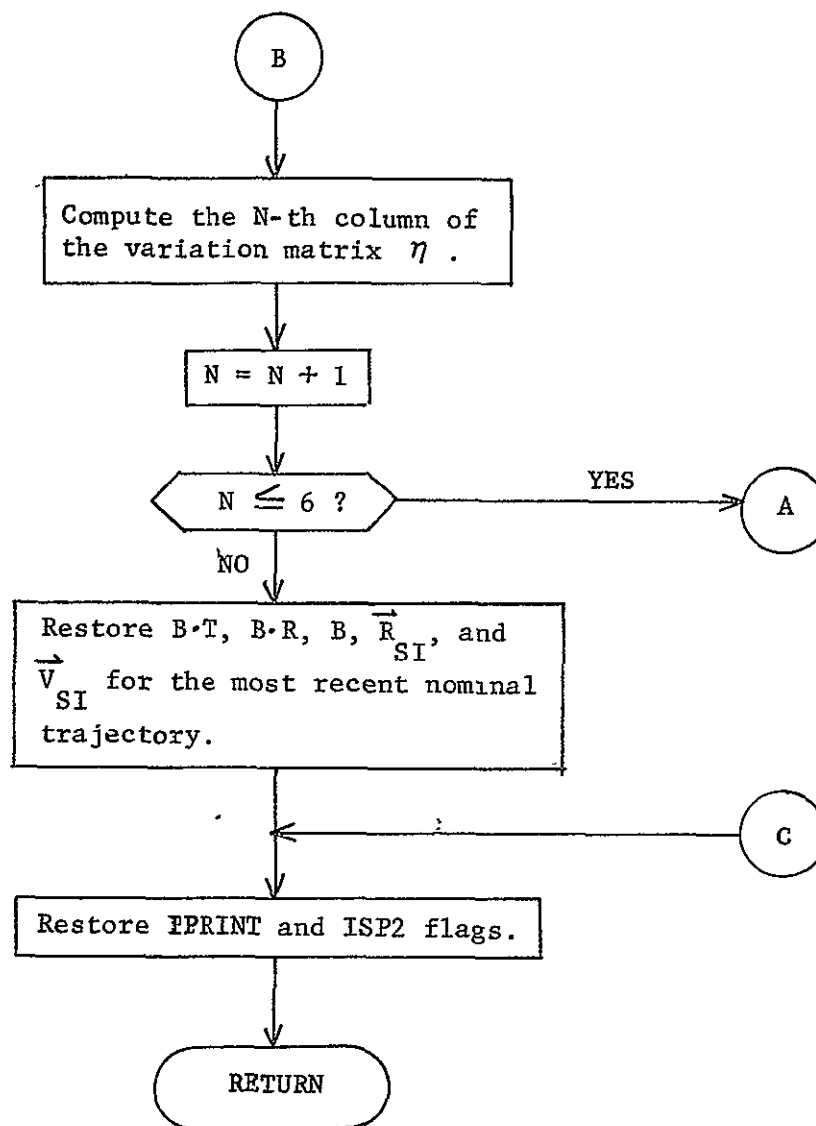
Let $\vec{\eta}_j$ be the j -th column of the matrix η , and assume (most recent) nominal $B \cdot T^*$, $B \cdot R^*$, t_{SI}^* , and \vec{X}_k^* are available. To obtain $\vec{\eta}_j$ we increment the j -th element of \vec{X}_k^* by the numerical differencing factor ΔX_j and numerically integrate the spacecraft equations of motion from t_k to the sphere of influence of the target planet to obtain the new values of $B \cdot T$, $B \cdot R$, and t_{SI} . Then

$$\vec{\eta}_j = \begin{bmatrix} \frac{B \cdot T - B \cdot T^*}{\Delta X_j} , & \frac{B \cdot R - B \cdot R^*}{\Delta X_j} , & \frac{t_{SI} - t_{SI}^*}{\Delta X_j} \end{bmatrix}^T$$

$$j = 1, 2, \dots, 6$$

VARSIM Flow Chart





SUBROUTINE VECTOR

PURPOSE: TO COMPUTE THE VECTOR ORBITAL ELEMENTS K (ANGULAR
MOMENTUM VECTOR), E (ECCENTRICITY VECTOR TOWARD
PERIHELION), TO COMPUTE THE SPACECRAFT FINAL POSITION
ON THE ORBIT TO ACCURATELY APPROXIMATE THE DESIRED TIME
INTERVAL, AND TO COMPUTE THE CONIC SECTION TIME OF
FLIGHT.

CALLING SEQUENCE: CALL VECTOR

SUBROUTINES SUPPORTED: VMP

SUBROUTINES REQUIRED: SPACE

LOCAL SYMBOLS: DUM INTERMEDIATE VARIABLE

COMMON COMPUTED/USED: V

COMMON COMPUTED: KOUNT

COMMON USED: HALF ITRAT ONE PI THREE
TWOPI TWO

VECTØR Analysis

The Kepler vector \vec{k} representing twice the areal rate of the spacecraft with respect to the virtual mass to be used during the current interval is computed from

$$\vec{k} = \vec{r}_{VS_B} \times \dot{\vec{r}}_{VS_B} \quad (1)$$

where the position and velocity vectors are referenced to the virtual mass at the beginning of the interval. The eccentricity vector for the interval is given by

$$\vec{e} = -\frac{\vec{r}_{VS_B}}{r_{VS_B}} - \frac{\vec{k} \times \vec{r}_{VS_B}}{\bar{\mu}_V} \quad (2)$$

where $\bar{\mu}_V$ is the average value of the virtual mass during the interval.

The current time interval is computed from

$$\Delta\tau = \Delta t + K \Delta t^2 \quad (3)$$

where the factor K was precomputed during the previous iterations. The direction of the final position $\vec{\sigma}$ is determined from

$$\vec{\sigma} = \vec{r}_{VS_B} + \Delta\tau \dot{\vec{r}}_{VS_B} \quad (4)$$

The magnitude factor B is chosen to force the final position to satisfy the orbit equation ($\vec{e} \cdot \vec{r} = -r + k^2/\mu$)

$$B = \frac{k^2/\bar{\mu}_V}{\vec{e} \cdot \vec{\sigma} + |\vec{\sigma}|} \quad (5)$$

The position and velocity vectors of the spacecraft relative to the virtual mass at the end of the interval are then

$$\begin{aligned} \vec{r}_{VS_E} &= B \vec{\sigma} \\ \dot{\vec{r}}_{VS_E} &= \frac{\bar{\mu}_V}{k^2} \left[\vec{k} \times \left(\vec{e} + \frac{\vec{r}_{VS_E}}{r_{VS_E}} \right) \right] \end{aligned} \quad (6)$$

The final position and velocity of the spacecraft in the reference inertial coordinates are computed from

$$\begin{aligned}\vec{r}_{S_E} &= \vec{r}_{VS_E} + \vec{r}_{V_E} \\ \dot{\vec{r}}_{S_E} &= \dot{\vec{r}}_{VS_E} + \dot{\vec{r}}_{V_E}\end{aligned}\quad (7)$$

The exact conic section time of flight is now computed. The in-plane normal to the major axis is

$$\begin{aligned}\vec{n} &= \frac{\vec{k} \times \vec{e}}{k e} & e \neq 0 \\ \frac{\vec{k} \times \vec{r}_{VS_B}}{k r_{VS_B}} & & e = 0\end{aligned}\quad (8)$$

The length of the semi-major axis is given by

$$\begin{aligned}a &= \frac{k^2}{\bar{\mu}_V |1-e^2|^{1/2}} & e \neq 1 \\ a_i &= \frac{2}{r_{VS_i} - k^2/\bar{\mu}_V} & e = 1, i = B, E\end{aligned}\quad (9)$$

The projection of the radius vector orthogonal to the major axis divided by a is given by

$$X_i = \frac{\vec{n} \cdot \vec{r}_{VS_i}}{a_i} \quad i = B, E \quad (10)$$

The mean angular rate is

$$\begin{aligned}\bar{\omega} &= \frac{\bar{\mu}_V (1-e^2)}{k a} & e \neq 1 \\ &= \frac{k}{2} & e = 1\end{aligned}\quad (11)$$

where $\omega < 0$ for hyperbolic orbits. The eccentric anomaly is given by

$$\begin{aligned}
 E_i &= \sin^{-1} X_i & e < 1 \\
 i=B,E \\
 &= \frac{k^2 / \bar{\mu}_V X_i}{3} & e = 1 \\
 &= \sinh^{-1} X_i & e > 1
 \end{aligned} \tag{12}$$

$$\text{Then } M_i = E_i - e X_i \quad i = B, E \tag{13}$$

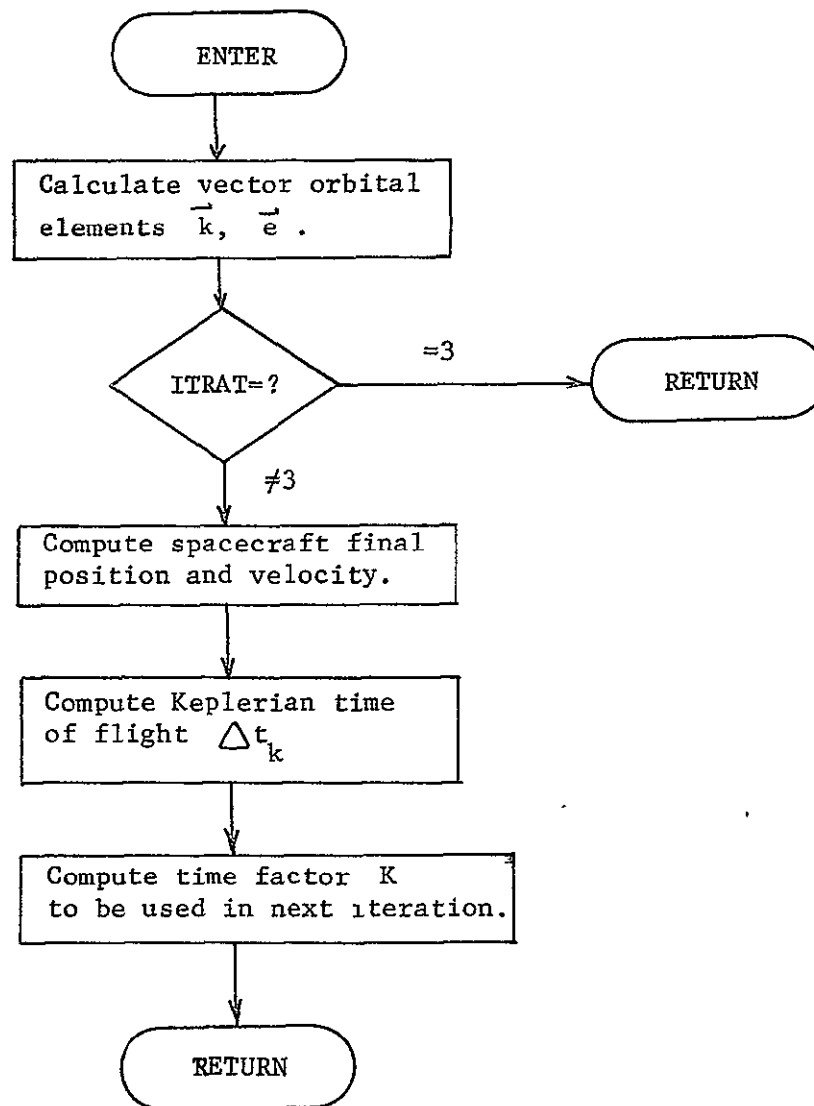
and the actual conic time of flight is

$$\Delta t = t_E - t_B = \frac{M_E - M_B}{\bar{\omega}} \tag{14}$$

The value of the time factor K to be used in the following interval is then computed

$$K = \frac{\Delta \tau - \Delta t}{(\Delta t)^2} \tag{15}$$

VECTOR Flow Chart



SUBROUTINE VMASS

PURPOSE: TO COMPUTE THE POSITION, VELOCITY, MAGNITUDE, AND
MAGNITUDE RATE OF THE VIRTUAL MASS.

CALLING SEQUENCE: CALL VMASS

SUBROUTINES SUPPORTED: VMP

SUBROUTINES REQUIRED: NONE

COMMON COMPUTED/USED: F V

COMMON USED: NBODY THREE ZERO

VMASS Analysis

The current virtual mass data is computed by VMASS. The magnitude and position of the virtual mass is given by

$$\mu_V = r_{VS}^3 M_S \quad (1)$$

$$\vec{r}_V = \frac{\vec{M}}{M_S} \quad (2)$$

where the intermediate variables are given by

$$\vec{M} = \sum_{i=1}^n \frac{\mu_i \vec{r}_i}{r_{iS}^3} \quad (3)$$

$$M_S = \sum_{i=1}^n \frac{\mu_i}{r_{iS}^3} \quad (4)$$

and of course $r_{iS} = |\vec{r}_i - \vec{r}_S|$ and $r_{VS} = |\vec{r}_V - \vec{r}_S|$ where \vec{r}_i represents the inertial position vector of the i -th body.

The time derivatives of these variables are given by

$$\dot{\mu}_V = \mu_V \left(\alpha_V + \frac{\dot{M}_S}{M} \right) \quad (5)$$

$$\dot{\vec{r}}_V = \frac{\dot{\vec{M}} - \vec{r}_V \dot{M}_S}{M_S} \quad (6)$$

$$\dot{\vec{M}} = \sum_{i=1}^n \frac{\mu_i}{r_{iS}^3} \left[\dot{\vec{r}}_i - \vec{r}_i \alpha_i \right] \quad (7)$$

$$\dot{M}_S = - \sum_{i=1}^n \frac{\mu_i}{r_{iS}^3} \alpha_{iS} \quad (8)$$

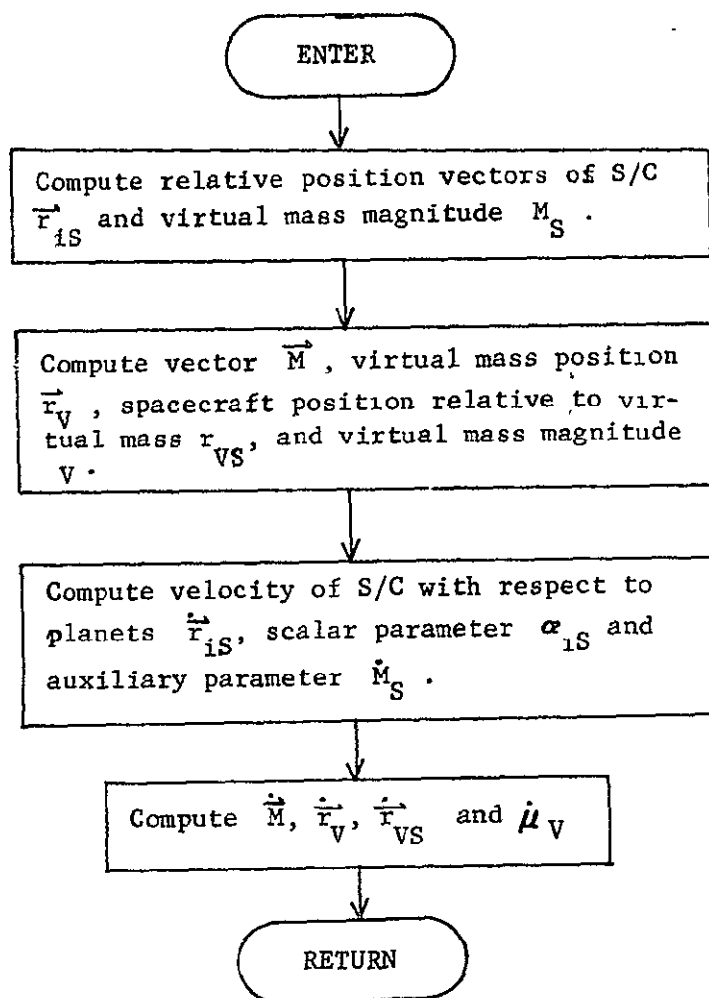
where

$$\alpha_{iS} = \frac{3 \vec{r}_{iS} \cdot \frac{d\vec{r}_{iS}}{dt}}{r_{iS}^2} \quad (9)$$

Finally, the velocity of the spacecraft with respect to the virtual mass is

$$\vec{r}_{VS} = \vec{r}_S - \vec{r}_V \quad (10)$$

VMASS Flow Chart



SUBROUTINE VMP

PURPOSE PROVIDE LOGIC TO GENERATE VIRTUAL MASS TRAJECTORY

CALLING SEQUENCE: CALL VMP(RS, ACC, D1, TRTM, DELTM, RSF, ISP2)

ARGUMENTS			
RS(6)	I	INERTIAL POSITION AND VELOCITY OF S/C AT INITIAL TIME	
ACC	I	ACCURACY USED IN INTEGRATION	
D1	I	JULIAN DATE, EPOCH 1900, OF INITIAL TIME	
TRTM	I	TRAJECTORY TIME (DAYS) AT INITIAL TIME	
DELTM	I	TIME INTERVAL IN DAYS OVER WHICH THE TRAJECTORY IS TO BE PROPAGATED UNLESS A STOPPING CONDITION IS REACHED	
RSF(6)	O	INERTIAL POSITION AND VELOCITY OF S/C AT FINAL TIME	
ISP2	I	SPHERE OF INFLUENCE STOPPING FLAG =0 DO NOT STOP AT SOI =1 STOP AT SOI IF INTERSECTED BEFORE FINAL TIME	

SUBROUTINES SUPPORTED: CASCAD NTMS GIDANS TARGET TARMAX
NTM TRJTRY DESENT
IPPROP IPRTRG MPPROP

SUBROUTINES REQUIRED:	CAREL	ELCAR	EPHEM	IMPACT	ORB
	PECEQ	TIME	ESTMT	INPUTZ	PRINT
	SPACE	VECTOR	VMASS		

LOCAL SYMBOLS	AU	NOT USED
	CXI	COSINE OF THE TRAJECTORY INCLINATION AT CLOSEST APPROACH
	D	INTERMEDIATE DATE FOR PRINTOUT PURPOSES
	DELR	INTERMEDIATE VARIABLE FOR INTERSECTION OF SPHERE-OF-INFLUENCE
	DELT	INTERMEDIATE TIME INCREMENT FOR INTERPOLATED SPHERE-OF-INFLUENCE POSITION
	ECEQP	TRANSFORMATION FROM ECLIPTIC TO EQUATORIAL SYSTEM FOR TARGET PLANET
	ICUT	CUTOFF FLAG USED WHEN CLOSEST APPROACH CUTOFF WAS DESIRED BUT NO VALID CLOSEST

APPROACH FOUND

IDAY	PRINTOUT CALENDAR DAY
IHR	PRINTOUT CALENDAR HOUR
IMO	PRINTOUT CALENDAR MONTH
IP	NUMBER OF PLANET, USED IN PRINTOUT
ISPHI	INDICATOR FOR CALCULATION OF SPECIAL COMPUTING INTERVAL NEAR TARGET PLANET SPHERE-OF-INFLUENCE
IYR	PRINTOUT CALENDAR YEAR
JJ	COUNTER FOR NUMBER OF ITERATIONS FOR INTERPOLATED SPHERE OF INFLUENCE
LARCA	INDICATOR FOR CALCULATION OF PSUEDO CLOSEST APPROACH
MIN	PRINTOUT CALENDAR MINUTES
NTPI	INDEX OF THE SPACECRAFT VECTORS IN THE F-ARRAY WITH RESPECT TO THE TARGET PLANET
RCM	MAGNITUDE OF POSITION OF VEHICLE RELATIVE TO TARGET PLANET AT CLOSEST APPROACH
RCM1	PREVIOUS RADIUS OF VEHICLE RELATIVE TO TARGET PLANET
RCM2	PRESENT RADIUS OF VEHICLE RELATIVE TO TARGET PLANET
ROD	NOT USED
RTEMP	SPACECRAFT POSITION AT INTERPOLATED CLOSEST APPROACH IN THE TARGET PLANET EQUATORIAL SYSTEM
SEC	PRINTOUT CALENDAR SECONDS
TIMCR	TIME INCREMENT USED FOR INTERPOLATED CLOSEST APPROACH
TIMIN	TOTAL TIME USED IN ONE INTEGRATED TRAJECTORY
TIM1	CLOCK TIME AT BEGINNING OF TRAJECTORY
TIM2	CLOCK TIME AT END OF TRAJECTORY

TMU GRAVITATIONAL CONSTANT OF TARGET PLANET
 (KM**3/SEC**2)
 TP INTERMEDIATE VARIABLE FOR CALCULATION OF
 SPECIAL COMPUTING INTERVAL NEAR SPHERE-OF-
 INFLUENCE OF TARGET PLANET
 ITG GRAVITATIONAL CONSTANT OF TARGET PLANET
 (KM**3/SEC**2)
 VCA VELOCITY MAGNITUDE WITH RESPECT TO TARGET
 PLANET AT INTERPOLATED CLOSEST APPROACH
 VCM MAGNITUDE OF VELOCITY OF VEHICLE RELATIVE
 TO TARGET PLANET AT CLOSEST APPROACH
 BEFORE INTERPOLATION
 VQ EQUATORIAL SPACECRAFT VELOCITY RELATIVE TO
 TARGET PLANET AT CLOSEST APPROACH BEFORE
 INTERPOLATION
 VTEMP SPACECRAFT VELOCITY AT INTERPOLATED
 CLOSEST APPROACH IN THE TARGET PLANET
 EQUATORIAL SYSTEM
 XI UNINTERPOLATED EQUATORIAL INCLINATION FOR
 PRINTOUT PURPOSES
 XMAG INTERMEDIATE VARIABLE FOR CALCULATION OF
 XI
 XN VECTOR NORMAL TO TRAJECTORY PLANE IN
 TARGET PLANET EQUATORIAL SYSTEM FOR
 CALCULATION OF XI
 XQ EQUATORIAL SPACECRAFT POSITION RELATIVE
 TO TARGET PLANET AT UNINTERPOLATED
 CLOSEST APPROACH
 ZM INTERMEDIATE VARIABLE IN CALCULATION OF
 INTERPOLATED CLOSEST APPROACH INCLINATION
 ZTEMP VECTOR NORMAL TO TRAJECTORY PLANE FOR
 CALCULATION OF INTERPOLATED CLOSEST
 APPROACH INCLINATION

COMMON COMPUTED/USED:	CAINC	DC	DSI	ICL	INCMNT
	INCMT	ISPH	ITRAT	KOUNT	NBODYI
	NO	RCA	RC	RSI	TIMINT
	VSI	V			

COMMON COMPUTED:	DELTH	INCPR	RE	RTP	RVS
------------------	-------	-------	----	-----	-----

VMU

COMMON USED:

ALNGTH	BDR	BDT	B	DELTP
EM7	EM8	F	HALF	ICL2
IEPHEM	INPR	IPRINT	NB00	NB
NTP	ONE	PLANET	PMASS	RADIUS
RAD	SPHERE	TM	TWO	ZERO

VMP Analysis

VMP provides the logic to integrate an N-body trajectory from an initial spacecraft state (\vec{r}_S, \vec{v}_S) at time t_B to one of the following stopping conditions.

1. Target planet sphere of influence (SOI) is reached (ISP2 \neq 0).
2. The closest approach to the target planet has been reached (ICL2 = 1).
3. The preset final trajectory time t_F has been exceeded.

The integration logic is controlled by ITRAT

- ITRAT = 1 First pass through computation cycle (including ephemeris computation).
- 2 Second pass through computation cycle (excluding ephemeris).
- 3 Initialization flag.

To start the integration, appropriate variables are initialized (PRINTZ) and ITRAT is set equal to 3. The state of all gravitational bodies at t_B are found (ORB, EPHEM). The initial virtual mass position \vec{r}_{V_B} , velocity \vec{v}_{V_B} , magnitude μ_{V_B} and magnitude rate $\dot{\mu}_{V_B}$ are found by VMAS. Virtual mass dependent values are then initialized

$$\mu_{V_{AVE}} = \mu_{V_E} = \mu_{V_B} \quad (1)$$

$$\dot{\vec{r}}_{V_{AVE}} = \dot{\vec{r}}_{V_B} \quad (2)$$

$$\vec{r}_{VS_E} = \vec{r}_{VS_B} \quad (3)$$

$$\dot{\vec{r}}_{VS_E} = \dot{\vec{r}}_{VS_B} \quad (4)$$

$$(\Delta t)^2 = 1 \quad (5)$$

$$ISPH1 = 0 \quad (6)$$

At this point the standard integration routine is entered by calling VECTOR.

In the standard integration routine, a new increment is initiated by calling ESTMT which:

1. Initializes all appropriate variables at the beginning of the increment (subscript B) to equal their values at the end of the previous increment.
2. Computes a Δt for the increment based on a modified true anomaly passage.
3. Computes the time at the end of the increment t_E .
4. Estimates the final (subscript E) position \vec{r}_{VE} and magnitude μ_{VE} of the virtual mass.

Based on these estimates, the average magnitude and velocity of the virtual mass is computed

$$\mu_{V\text{AVE}} = 1/2 (\mu_{VB} + \mu_{VE}) \quad (7)$$

$$\vec{v}_{V\text{AVE}} = (\vec{r}_{VE} - \vec{r}_{VB})/\Delta t \quad (8)$$

Subroutine VECTOR then computes the orbit relative to the virtual mass based on these estimates. It also refines the estimate of the spacecraft final state $(\vec{r}_{SE}, \vec{v}_{SE})$. ORB and EPHEM are called to determine the state at t_E of all gravitational bodies being considered. The virtual mass position \vec{r}_{VE} , velocity \vec{v}_{VE} , magnitude μ_{VE} and magnitude rate $\dot{\mu}_{VE}$ are determined by VMAS.

Using these refined values, the virtual mass average magnitude $\mu_{V\text{AVE}}$ and velocity $\vec{v}_{V\text{AVE}}$ are recomputed using equations (7) and (8). At this point a second pass is made through VECTOR to compute the spacecraft final state $(\vec{r}_{SE}, \vec{v}_{SE})$ which will be used in all subsequent calculations. VMAS is again called to make a final determination of the virtual mass

position, velocity, magnitude and magnitude rate at the end of the increment.

The virtual mass average accelerations are then computed

$$\ddot{\mu}_{V_{AVE}} = \left[\mu_{V_E} - \mu_{V_B} - \dot{\mu}_{V_B} (\Delta t) \right] / (\Delta t)^2 \quad (9)$$

$$\dot{\vec{v}}_{V_{AVE}} = \left[\vec{r}_{V_E} - \vec{r}_{V_B} - \vec{v}_{V_B} (\Delta t) \right] / (\Delta t)^2 \quad (10)$$

These values are subsequently used by ESTMT to estimate the final position \vec{r}_{VE} and magnitude μ_{VE} of the virtual mass for the next increment.

Tests are now made to determine whether the vehicle is inside a planetocentric sphere of radius 1.025 times larger than that of the SØI. If it is not, integration goes on as usual. If it is and yet is still outside the SØI, the integration step size is reduced to obtain an integration state near enough the SØI to permit accurate extrapolation to it. Finally, if the vehicle is inside of the sphere of influence, a refined sphere of influence (SØI) state is constructed by fitting an osculating planetocentric conic to the current state and extrapolating it to the sphere. The entire refinement process is carried out in subroutine SØIPS.

The refined state at the SØI is then used by IMPACT to compute B•T and B•R.

If trajectory data are to be printed at this point, the orbit inclination (assuming a hyperbolic orbit about the planet) is computed by first determining the "Kepler vector"

$$\vec{k} = \vec{r}_{ST} \times \vec{v}_{ST} \quad (11)$$

in planetocentric equatorial coordinates. Then

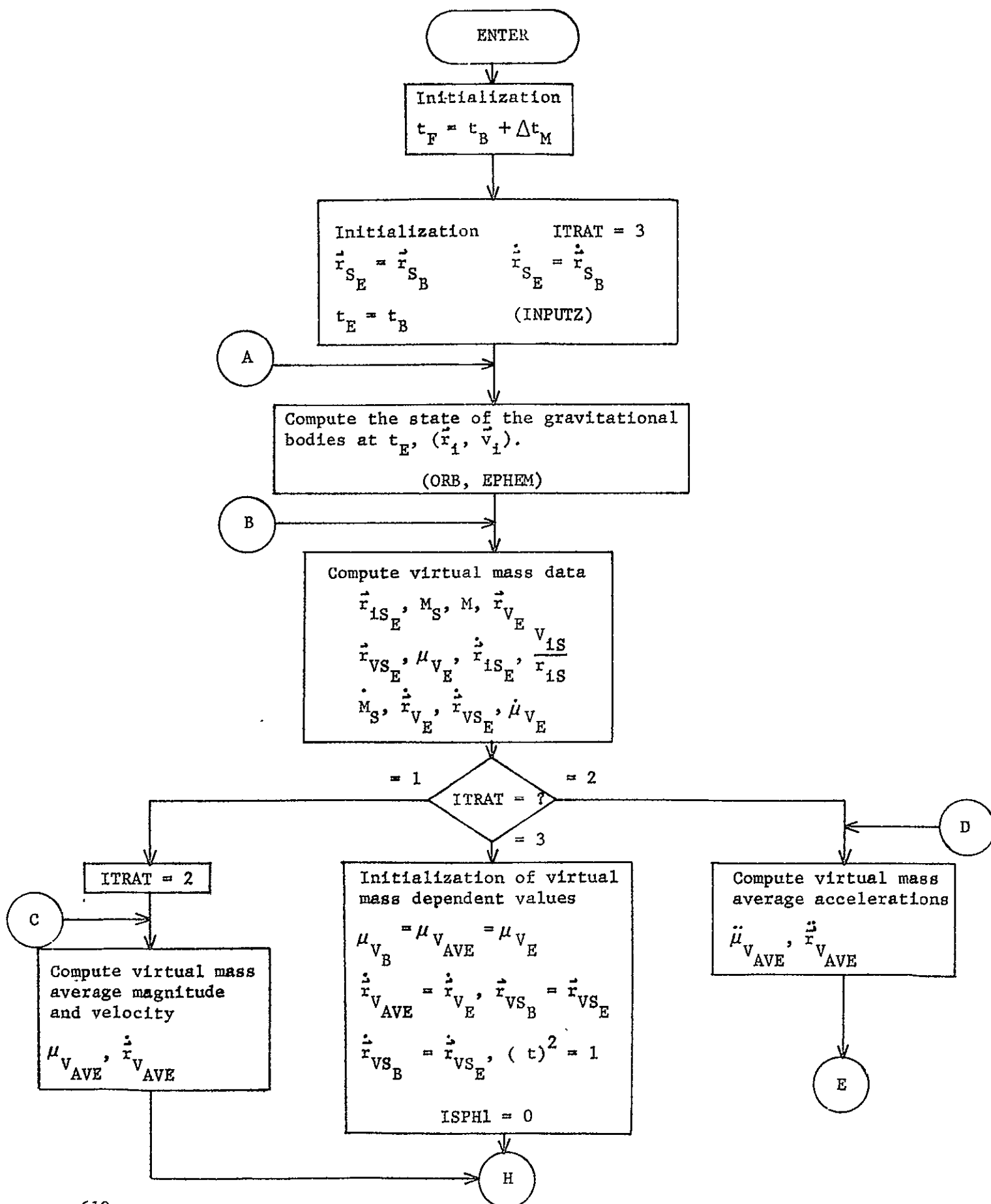
$$\cos i = \frac{k_z}{|\vec{k}|} \quad (12)$$

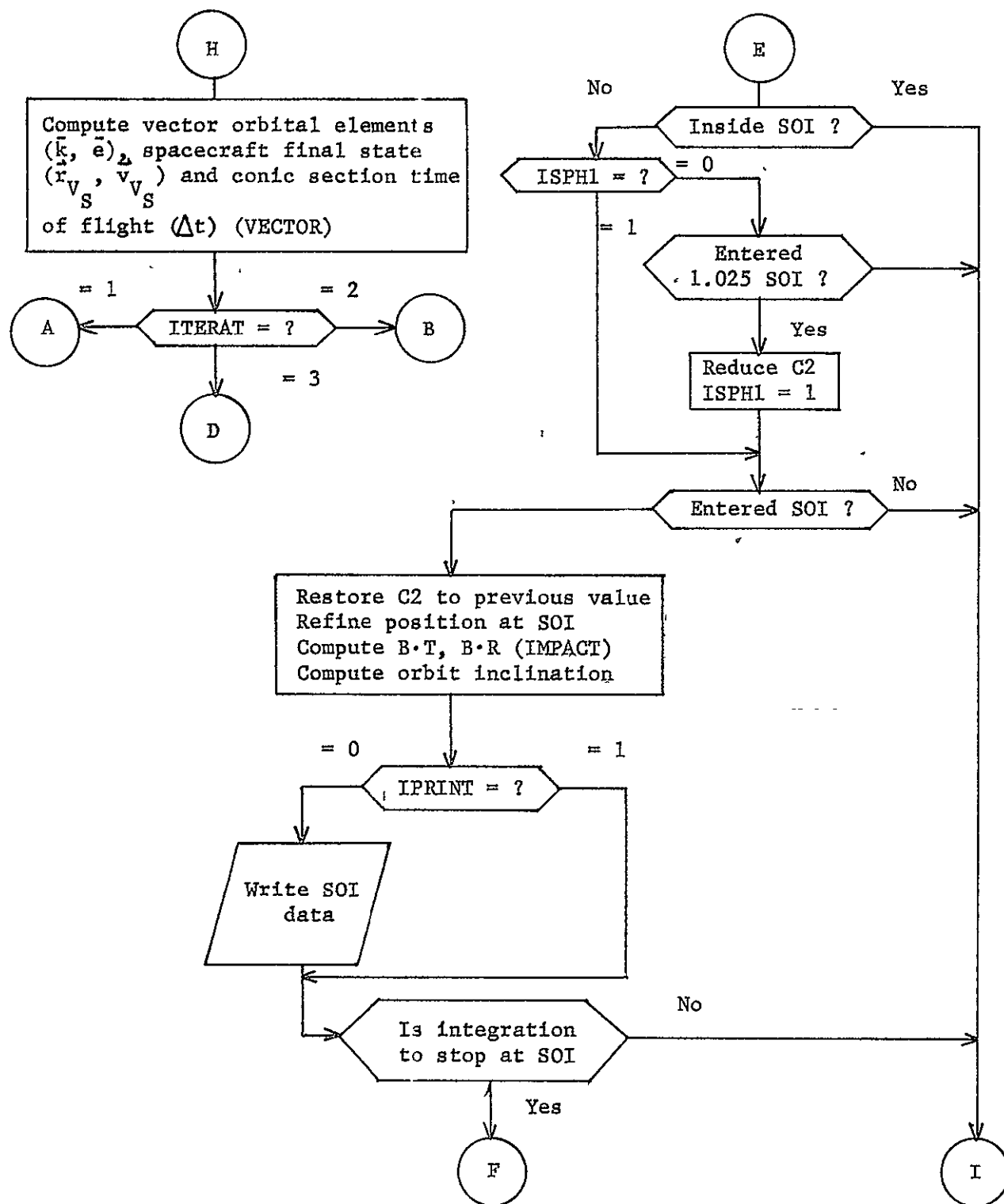
where i = orbit inclination and \vec{k}_z = component of \vec{k} normal to planet equatorial plane.

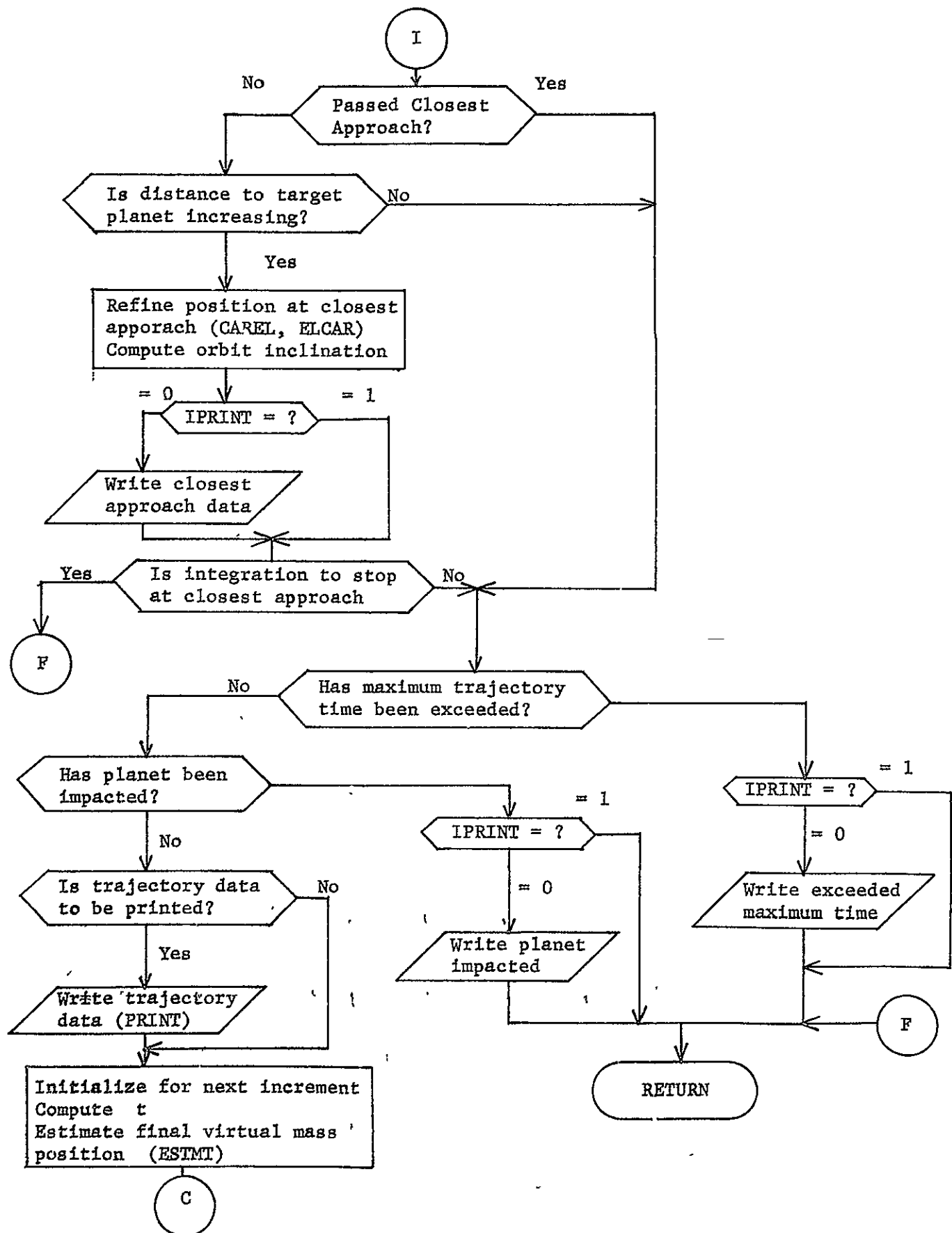
Tests are now made to determine if the spacecraft has reached a closest approach to the target planet. If it has, the interpolated state at closest approach $(\bar{r}_{CA}, \bar{v}_{CA})$ is computed by calling CAREL with the spacecraft state just following closest approach. CAREL returns the element of the near planet conic. ELCAR is then called with these conic elements and returns the interpolated state at closest approach.

If the spacecraft is not within 10 SOI of the target planet, printout of closest approach data may occur; however, integration continues.

The final tests before starting a new integration increment determine if the maximum trajectory time t_F has been exceeded or a planet has been impacted. If the latter has occurred, the actual impact state is determined by fitting an osculating planetocentric conic to the current state and extrapolating to the planet surface. As was the case earlier at the SOI, this procedure is carried out in the subroutine SØIPS. If the impacted planet is the object of any type of probe targeting, the planet radius used in the impact state computation is the probe-sphere value input by the user. If these final two tests are passed, a new integration cycle is initiated by calling ESTMT.







SUBROUTINE ZERIT

PURPOSE: TO COMPUTE THE COMPUTATION OF THE ZERO ITERATE VALUES OF
TIME, POSITION VECTOR, AND VELOCITY VECTOR.

CALLING SEQUENCE: CALL ZERIT

SUBROUTINES SUPPORTED: PRELIM GIDANS

SUBROUTINES REQUIRED: HELIO LUNA

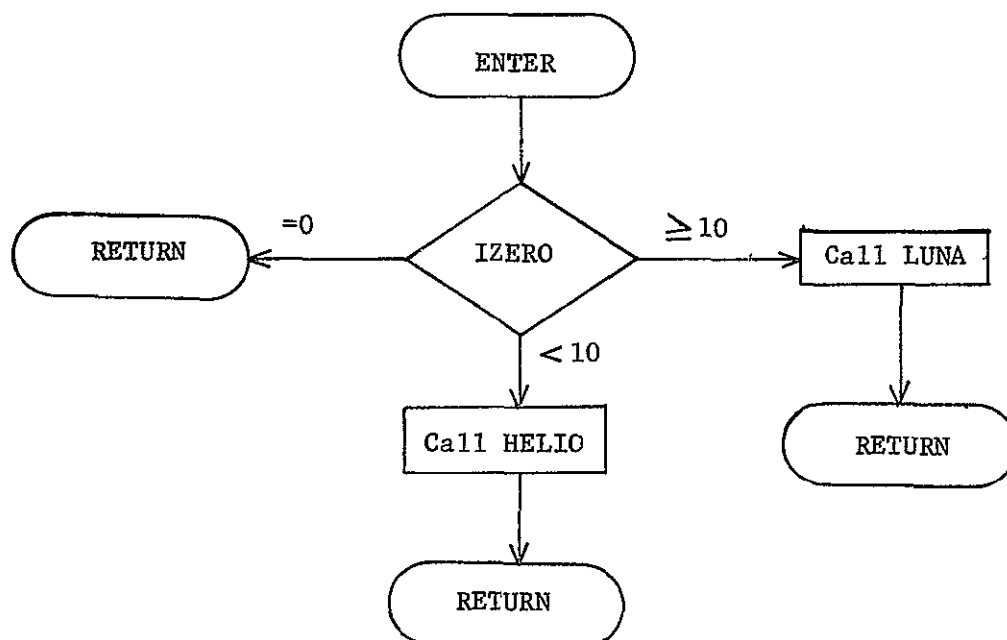
COMMON USED: IZERO LTARG

ZERIT Analysis

ZERIT is the executive subroutine handling the computation of the zero iterate values of time, position vector, and velocity vector.

The flag IZERO controls the operation of ZERIT. If $IZERO = 0$, no zero iterate computation is needed and so ZERIT is exited. If $IZERO < 10$, the zero iterate is to be computed for an interplanetary trajectory so HELIO is called before returning. If $IZERO \geq 10$, the zero iterate is to be computed for a lunar trajectory so LUNA is called for that computation.

ZERIT Flow Chart



SUBROUTINE ZTRANS

PURPOSE: TO CALCULATE THE TRANSCENDENTAL FUNCTIONS USED IN THE
 UNIVERSAL FORM OF KEPLER-S EQUATION

CALLING SEQUENCE: CALL ZTRANS(X,S,C)

ARGUMENTS: X I ANGLE
 S O SIN OR SINH OF X
 C O COS OR COSH OF X

SUBROUTINES SUPPORTED: BATCON

LOCAL SYMBOLS: CH INTERMEDIATE VARIABLE
 SH INTERMEDIATE VARIABLE
 T1 INTERMEDIATE VARIABLE
 T2 INTERMEDIATE VARIABLE
 Y INTERMEDIATE VARIABLE

BIBLIOGRAPHY

1. Baker, R.M.L. and Makemson, M.W. : An Introduction to Astrodynamics, Academic Press, New York, 1967.
2. Battin, R.H. : Astronautical Guidance, McGraw-Hill Book Company, Inc., New York, 1964.
3. Battin, R.H. and Fraser, D.C. : Space Guidance and Navigation, AIAA Professional Study Series, 1970.
4. Byrnes, D.V. and Hooper, H.L. : Multiconic: A Fast and Accurate Method of Computing Space Flight Trajectories, AAS/AIAA Astrodynamics Conference, Santa Barbara, Cal., 1970, AIAA Paper 70-1062.
5. Danby, J.M.A. : Matrizant of Keplerian Motion, AIAA J., vol 3, no. 4, April, 1965.
6. Hoffman, L.H. and Young, G.R. : Approximation to the Statistics of Midcourse Velocity Corrections, NASA TN D-5381, Clearinghouse for Federal Scientific and Technical Information, Springfield, Virginia.
7. Jazwinski, A.H. : Adaptive Filtering, Analytical Mechanics Associates, Lanham, Maryland, May, 1968.
8. Joseph, A.E. and Richard, R.J. : Space Research Conic Program, Engineering and Planning Document 406, Jet Propulsion Laboratory, July, 1966.
9. Kizner, W. : A Method of Describing Miss Distances for Lunar and Interplanetary Trajectories Under the Influence of Multiple Planet Attractions, TR-32-464, Jet Propulsion Laboratory, 1963.
10. Lee, B.G., Falce, R.R., and Hopper, F. : Interplanetary Trajectory Error Analysis, Volume I - Analytical Manual, Vol II - Computer Program Design, MCR-67-441, Martin-Marietta Corporation, Denver, Colorado, December, 1967. (Completed under Contract NAS8-21120)
11. Lee, B.G., Vogt, E.D., Falce, R.R., Pearson, S., and Demlow, E. : Simulated Trajectories Error Analysis Program, Vol I - Users Manual, Vol II - Analytical Manual, NASA CR-66818, Martin Marietta Corporation, Denver, Colorado. (Completed under Contract NAS1-8745)
12. Lesh, H.F. and Travis, C. : FLIGHT: A Subroutine to Solve the Flight Time Problem, JPL Space Programs Summary 37-53, Vol II.

Preceding page blank

13. Mitchell, R.T. and Wong, S.K. : Preliminary Flight Path Analysis, Orbit Determination and Maneuver Strategy, Mariner Mars 1969, Project Document 138, Jet Propulsion Laboratory, 1968.
14. Myers, G.E. : Properties of the Conjugate Gradient and Davidon Methods, AAS Paper 68-081. Presented at 1968 AAS/AIAA Astrodynamics Specialist Conference, Jackson, Wyoming.
15. Novak, D.H. : Virtual Mass Technique for Computing Space Trajectories, ER-14045, Martin-Marietta Corporation, Denver, Colorado, January 1966. (Completed under Contract NAS9-4370)
16. Scheid, F. : Theory and Problems of Numerical Analysis, McGraw-Hill Book Company, Inc., New York, 1968.
17. Sorenson, H. : Kalman Filtering, Advances in Control Systems, vol. 3, C.T. Leades (Ed.), Academic Press, New York, 1966.